Hackers

Steven Levy

Sobre el Autor

Levy es redactor senior de la revista Wired. Anteriormente, fue redactor jefe de tecnología y editor senior de Newsweek.

Levy ha escrito seis libros y ha publicado artículos en Harper's, Macworld, The New York Times Magazine, The New Yorker, Premiere y Rolling Stone.

Steven ha ganado varios premios durante sus más de 30 años de escritura sobre tecnología, entre ellos *Hackers* - que PC Magazine nombró el mejor libro de ciencia y tecnología escrito en los últimos veinte años - y *Crypto*, que ganó el gran premio de libros electrónicos en el Festival del Libro de Frankfurt de 2001.

Colofón

Las tipografías de la tapa son Berthold Akzidenz Grotesk y Helvetica Neue. La tipografía del texto es Sabon; la tipografía del encabezado es Myriad.

Editor

Mike Hendrickson

Copyright © 2010 Steven Levy

Los libros de O’Reilly se pueden comprar para uso educativo, comercial o promocional. También hay ediciones en línea disponibles para la mayoría de los títulos (http://my.safaribooksonline.com). Para obtener más información, comuníquese con nuestro departamento de ventas corporativo/institucional: (800) 998-9938 o corporate@oreilly.com.

El logotipo de O’Reilly es una marca registrada de O’Reilly Media, Inc. Hackers y la imagen comercial relacionada son marcas registradas de O’Reilly Media, Inc.

Muchas de las designaciones utilizadas por los fabricantes y vendedores para distinguir sus productos se reclaman como marcas registradas. Cuando esas designaciones aparecen en este libro, y O’Reilly Media, Inc. tenía conocimiento de una reclamación de marca registrada, las designaciones se han impreso en mayúsculas o mayúsculas iniciales. Si bien se han tomado todas las precauciones en la preparación de este libro, el editor y los autores no asumen ninguna responsabilidad por errores u omisiones, o por daños resultantes del uso de la información contenida en este documento.

Prefacio

Al principio me sentí atraído a escribir sobre los hackers (esos programadores y diseñadores informáticos que consideran que la informática es lo más importante del mundo) porque eran personas fascinantes. Aunque algunos en el campo usaban el término “hacker” como una forma de burla - dando a entender que los hackers eran parias sociales nerds o programadores “poco profesionales” que escribían código informático sucio y “no estándar” - a mí me parecieron bastante diferentes. Debajo de sus fachadas a menudo poco imponentes, eran aventureros, visionarios, audaces, artistas… y los que veían con más claridad por qué la computadora era una herramienta verdaderamente revolucionaria. Entre ellos, sabían hasta dónde se podía llegar sumergiéndose en la concentración profunda de la mentalidad del hacker: se podía llegar infinitamente lejos. Llegué a entender por qué los verdaderos hackers consideran el término un apelativo honorífico en lugar de peyorativo.

Mientras hablaba con estos exploradores digitales - desde aquellos que domesticaron máquinas multimillonarias en los años 50 hasta los jóvenes magos contemporáneos que dominaron las computadoras en sus dormitorios suburbanos - encontré un elemento común, una filosofía común que parecía vinculada a la lógica elegantemente fluida de la computadora misma. Era una filosofía de compartir, apertura, descentralización y de tener las manos en las máquinas a cualquier costo para mejorar las máquinas y mejorar el mundo. Esta *ética hacker* ha sido su regalo para nosotros: algo con valor incluso para aquellos de nosotros que no tenemos ningún interés en las computadoras. Es una ética rara vez codificada, sino mas bien encarnada en el comportamiento de los propios hackers. Me gustaría presentarles a estas personas que no solo vieron, sino que vivieron la magia de la computadora y trabajaron para liberar la magia para que pudiera beneficiarnos a todos. Estas personas incluyen a los verdaderos hackers del Laboratorio de Inteligencia Artificial del MIT en los años cincuenta y sesenta; los hackers de hardware populistas y menos aislados de California en los años setenta; y los jóvenes hackers de videojuegos que dejaron su huella en la computación personal de los años ochenta.

Esta no es de ninguna manera una historia formal en los que me centro en la era del cómputo, o de los ámbitos particulares. De hecho, muchas de las personas que conocerá aquí no son los nombres más famosos en los anales de la informática (ciertamente no los más ricos). En vez de aquellos, estos son los genios en la trastienda, que entendieron la máquina en sus niveles más profundos y nos presentaron un nuevo tipo de estilo de vida y un nuevo tipo de héroe.

Los hackers como Richard Greenblatt, Bill Gosper, Lee Felsenstein y John Harris son el espíritu y el alma del cómputo en sí. Creo que su historia —su visión, su intimidad con la máquina en sí, sus experiencias dentro de su mundo peculiar y sus "interfaces" a veces dramáticas, a veces absurdas, con el mundo exterior— es la verdadera historia de la revolución informática.

Quién es quién: los magos y sus máquinas

Bob Albrecht. Fundador de People's Computer Company, que disfrutaba visceralmente de exponer a los jóvenes a las computadoras.

Altair 8800. La microcomputadora pionera que galvanizó a los hackers de hardware. Construir este kit te hacía aprender a hackear. Luego intentabas averiguar qué hacer con él.

Apple II. La computadora amigable, escamosa y atractiva de Steve Wozniak, tremendamente exitosa y la chispa y el alma de una industria próspera.

Atari 800. Esta computadora doméstica proporcionó excelentes gráficos a los hackers de juegos como John Harris, aunque la compañía que la fabricó se resistía a decirte cómo funcionaba.

Bob y Carolyn Box. Buscadores de oro con récords mundiales que se convirtieron en estrellas del software y trabajaron para Sierra On-Line.

Doug Carlston. Abogado corporativo que lo dejó todo para formar la compañía de software Brøderbund.

Bob Davis. Dejó un trabajo en una licorería para convertirse en el autor más vendido del juego de computadora de Sierra On-Line, Ulises y el vellocino de oro. El éxito fue su perdición.

Peter Deutsch. Malo en deportes, brillante en matemáticas, Peter todavía llevaba pantalones cortos cuando se topó con el TX-0 en el MIT y lo hackeó junto con los maestros.

Steve Dompier. Miembro del Homebrew que primero hizo cantar a Altair y luego escribió el juego Target en la Sun, que cautivó a Tom Snyder.

John Draper. El famoso "Capitán Crunch" que exploró sin miedo los sistemas telefónicos, fue encarcelado y luego hackeó microcomputadoras. Los cigarrillos lo volvieron violento.

Mark Duchaineau. El joven Dungeonmaster que protegía contra copias los discos de On-Line a su antojo.

Chris Espinosa. Seguidor de Steve Wozniak de catorce años y primer empleado de Apple.

Lee Felsenstein. Ex "editor militante" del Berkeley Barb y héroe de una novela imaginaria de ciencia ficción, diseñó computadoras con un enfoque de "depósito de chatarra" y fue una figura central en el hackeo de hardware del Área de la Bahía en los años setenta.

Ed Fredkin. Gentil fundador de Information International, se consideraba el mejor programador del mundo hasta que conoció a Stew Nelson. Figura paterna para los hackers.

Gordon French. Hacker de hardware de cabello plateado cuyo garaje no tenía autos sino su computadora Chicken Hawk de fabricación casera, y luego celebró la primera reunión del Homebrew Computer Club.

Richard Garriott. Hijo de un astronauta que, bajo el sosias de Lord British, creó el mundo Ultima en discos de computadora.

Bill Gates. Mago arrogante y desertor de Harvard que escribió Altair BASIC y se quejó cuando los hackers lo copiaron.

Bill Gosper. Horowitz de los teclados de computadora, maestro de las matemáticas y hacker de LIFE en el laboratorio de IA del MIT, gurú de la ética hacker y estudiante de los menús de los restaurantes chinos.

Richard Greenblatt. Hacker del MIT, de mente cerrada, desaliñado, prolífico y canónico, que entró en la fase nocturna con tanta frecuencia que arruinó su carrera académica. El hacker entre los hackers.

John Harris. El joven hacker de juegos de Atari 800 que se convirtió en el programador estrella de Sierra On-Line, pero anhelaba la compañía femenina.

IBM PC. La entrada de IBM al mercado de las microcomputadoras, que sorprendentemente incluía un poco de la ética hacker y tomó el control.

IBM 704. IBM era el enemigo y esta era su máquina, la enorme computadora gigante en el edificio 26 del MIT. Más tarde modificada en la IBM 709, luego en la IBM 7090. Procesaba por lotes, lo cual la convertía en intolerable.

Jerry Jewell. Veterano de Vietnam convertido en programador que fundó Sirius Software.

Steven Jobs. Joven visionario, con cuentas y no hacker que tomó la Apple II de Wozniak, hizo muchos tratos y formó una compañía que ganaría mil millones de dólares.

Tom Knight. A los dieciséis años, un hacker del MIT que daría nombre al Sistema de Tiempo Compartido Incompatible. Más tarde, un némesis de Greenblatt en el cisma de las máquinas LISP.

Alan Kotok. El rechoncho estudiante del MIT de Jersey que trabajó bajo el diseño ferroviario en el TMRC, aprendió el sistema telefónico en Western Electric y se convirtió en un legendario hacker de TX-0 y PDP-1.

Efrem Lipkin. Activista hacker de Nueva York que amaba las máquinas pero odiaba sus usos. Cofundador de Community Memory; amigo de Felsenstein.

Máquina LISP. La computadora hacker definitiva, inventada principalmente por Greenblatt y objeto de una amarga disputa en el MIT.

"Tío" John McCarthy. Profesor distraído pero brillante del MIT (más tarde Stanford) que ayudó a ser pionero en ajedrez informático, inteligencia artificial y LISP.

Bob Marsh. Berkeleyiano y computador casero que compartió garaje con Felsenstein y fundó Processor Technology, que fabricó la computadora Sun.

Roger Melen. Computador casero que cofundó la empresa Cromemco para fabricar placas de circuitos para Altair. Su programa “Dazzler” reproducía LIFE en la mesa de su cocina.

Louis Merton. Seudónimo del hacker de ajedrez de IA cuya tendencia a volverse catatónico unió a la comunidad hacker.

Jude Milhon. Conoció a Lee Felsenstein a través de un anuncio clasificado en el Berkeley Barb y se convirtió en algo más que una amiga: un miembro del colectivo Community Memory.

Marvin Minsky. Profesor del MIT, juguetón y brillante, que dirigió el laboratorio de IA y permitió que los hackers actuaran libremente.

Fred Moore. Pacifista vagabundo que odiaba el dinero, amaba la tecnología y cofundó el Homebrew Club.

Stewart Nelson. Hacker de laboratorio de IA, diminuto, pero fogoso, que conectó la computadora PDP-1 para hackear el sistema telefónico. Más tarde cofundó la empresa Systems Concepts.

Ted Nelson. Autodenominado “innovador” y conocido cascarrabias que publicó por su cuenta el influyente libro Computer Lib.

Russell Noftsker. Acosado administrador del laboratorio de IA del MIT a finales de los sesenta; más tarde presidente de la empresa Symbolics.

Adam Osborne. Editor nacido en Bangkok convertido en fabricante de ordenadores que se consideraba un filósofo. Fundó la Osborne Computer Company para fabricar máquinas “adecuadas”.

PDP-1. El primer miniordenador de Digital Equipment y, en 1961, una bendición interactiva para los hackers del MIT y una bofetada en la cara al fascismo de IBM.

PDP-6. Diseñado en parte por Kotok, este mainframe fue la piedra angular del laboratorio de IA, con su magnífico conjunto de instrucciones y dieciséis registros sexys.

Tom Pittman. El hacker religioso de Homebrew que perdió a su mujer pero mantuvo la fe con su pequeño BASIC.

Ed Roberts. Enigmático fundador de la compañía MITS que sacudió al mundo con su computadora Altair. Quería ayudar a la gente a construir pirámides mentales.

Steve (Slug) Russell. El "coolie" de McCarthy que hackeó el programa Spacewar, el primer videojuego, en el PDP-1. Nunca ganó un centavo con eso.

Peter Samson. Hacker del MIT (uno de los primeros), que amaba los sistemas, los trenes, TX-0, la música, el procedimiento parlamentario, las bromas y el hackeo.

Bob Saunders. Hacker alegre y calvo de TMRC que se casó joven, hackeó hasta altas horas de la noche comiendo "lemon gunkies" y dominó la "estrategia de CBS" en Spacewar.

Warren Schwader. Hacker rubio y corpulento de la zona rural de Wisconsin que pasó de la línea de ensamblaje al estrellato del software, pero no pudo reconciliar el cambio con su devoción a los Testigos de Jehová.

David Silver. Dejó la escuela a los catorce años para ser la mascota del laboratorio de IA; fabricante de claves ilícitas y constructor de un pequeño robot que hacía lo imposible.

Dan Sokol. Bromista de pelo largo que se deleitaba revelando secretos tecnológicos en el Homebrew Club. Ayudó a "liberar" el programa BASIC de Altair en cinta de papel.

Sun Computer. La terminal y computadora de Lee Felsenstein, construida en dos meses frenéticos, casi la computadora que cambió las cosas. Casi no fue suficiente.

Les Solomon. Editor de Popular Electronics, el marionetista que puso en marcha la revolución informática.

Marty Spergel. El hombre de la chatarra, el miembro de Homebrew que suministraba circuitos y cables y podía hacerte un trato por cualquier cosa.

Richard Stallman. El último de los hackers, juró defender los principios del hackerismo hasta el final. Se quedó en el MIT hasta que no hubo nadie con quien comer comida china.

Jeff Stephenson. Veterano de las artes marciales y hacker de treinta años que se quedó atónito al saber que unirse a Sierra On-Line significaba inscribirse en un campamento de verano.

Jay Sullivan. Programador de nivel mago en Informática, increíblemente tranquilo, que impresionó a Ken Williams al saber el significado de la palabra "cualquiera".

Dick Sunderland. MBA de tez tiza que creía que la burocracia gerencial firme era un objetivo digno, pero como presidente de Sierra On-Line descubrió que los hackers no pensaban de esa manera.

Gerry Sussman. Joven hacker del MIT tildado de "perdedor" porque fumaba en pipa y "destruía" sus programas; más tarde se convirtió en "ganador" por su magia algorítmica.

Margot Tommervik. Con su marido Al, la melenuda Margot convirtió sus ganancias en concursos en una revista que deificó a Apple Computer.

Tom Swift Terminal. La legendaria terminal de computadora de Lee Felsenstein, que nunca se construyó, le daría al usuario la máxima libertad para poner sus manos sobre el mundo.

TX-0. Ocupaba una pequeña habitación, pero a finales de los años cincuenta, esta máquina de 3 millones de dólares, para la comunidad de hackers del MIT que se formó a su alrededor, fue la primera computadora personal del mundo.

Jim Warren. Repartidor corpulento de "tecno-chismes" en Homebrew, fue el primer editor del Dr. Dobbs Journal, de estilo hippie, y más tarde fundó la lucrativa Computer Faire.

Randy Wigginton. Miembro de quince años del grupo de niños de Steve Wozniak, ayudó a Woz a llevar la Apple II a Homebrew. Todavía estaba en la escuela secundaria cuando se convirtió en el primer empleado de software de Apple.

Ken Williams. Joven programador arrogante y brillante que vio lo que estaba escrito en el CRT y comenzó Sierra On-Line para ganar dinero y mejorar la sociedad vendiendo juegos para la computadora Apple.

Roberta Williams. La tímida esposa de Ken Williams, que redescubrió su propia creatividad al escribir Mystery House, el primero de sus muchos juegos de computadora multiventas.

Stephen “Woz” Wozniak. Un hacker de hardware de corazón abierto y tecnológicamente atrevido de los suburbios de San José, Woz construyó la computadora Apple para el placer de él mismo y sus amigos.

Parte I. Verdaderos Hackers: Cambridge: Los Cincuentas y Sesentas

Capítulo 1: El Club de Ferromodelismo

El motivo por el cual Peter Samson deambulaba por el edificio 26 en mitad de la noche es algo que le resultaría difícil de explicar. Mejor no hablar de ciertas cosas. Si usted fuera como la gente a la que Peter Samson llegó a conocer y de la que se hizo amigo en el invierno de 1958-59 - su primer año en el Instituto Tecnológico de Massachusetts – no le haría falta ninguna explicación. Deambular por el laberinto de laboratorios y almacenes, explorando los secretos de la conmutación telefónica en las salas de máquinas, siguiendo las rutas de los cables o relés en los túneles subterráneos de vapor... para algunos, era un comportamiento habitual y no había necesidad de justificar el impulso, cuando se enfrentaban a una puerta cerrada que escondía un ruido insoportablemente intrigante, de abrirla sin invitación. Y luego - si no había nadie que impidiera físicamente el acceso a lo que fuera que estaba haciendo ese ruido intrigante - tocar la máquina, empezar a accionar interruptores y a anotar respuestas, y finalmente aflojar un tornillo, desatornillar una caja, mover algunos diodos y ajustar algunas conexiones. Peter Samson y sus amigos habían crecido con una relación específica con el mundo, en la que las cosas tenían significado solo si descubrías cómo funcionaban. ¿Y cómo lo harías si no era poniéndoles las manos encima?

Fue en el sótano del Edificio 26 donde Samson y sus amigos descubrieron la sala EAM. El Edificio 26 era una larga estructura de vidrio y acero, uno de los edificios más nuevos del MIT, que contrastaba con las venerables estructuras de columnas que se encontraban frente al Instituto en Massachusetts Avenue. En el sótano de este edificio carente de personalidad, la sala EAM. *Maquinaria de Contabilidad Electrónica*. Una sala que albergaba máquinas que funcionaban como computadoras.

No mucha gente había visto siquiera una computadora en 1959, y mucho menos tocado una. Samson, un pelirrojo fibroso y rizado que alargaba las vocales de forma que parecía que estuviera repasando rápidamente listas de posibles significados de enunciados a mitad de palabra, había visto ordenadores en sus visitas al MIT desde su ciudad natal de Lowell, Massachusetts, a menos de cincuenta kilómetros del campus. Esto lo convertía en un "escozor de Cambridge", uno de las docenas de estudiantes de secundaria de la región apasionados por la ciencia que se sentían atraídos - como por una fuerza gravitacional - al campus de Cambridge. Incluso había intentado armar su propio ordenador con piezas desechadas de viejas máquinas de pinball: eran la mejor fuente de elementos lógicos que pudo encontrar.

ELEMENTOS LÓGICOS: el término parece resumir lo que atrajo a Peter Samson - hijo de un reparador de maquinaria de fábrica - a la electrónica. El tema tenía sentido. Cuando creces con una curiosidad insaciable por saber cómo funcionan las cosas, el deleite que sientes al descubrir algo tan elegante como la lógica de circuitos - donde todas las conexiones tienen que completar sus bucles - es profundamente emocionante. Peter Samson, que desde muy temprano apreció la simplicidad matemática de estas cosas, recordaba haber visto un programa de televisión en el canal público de Boston – WGBH - que ofrecía una introducción rudimentaria a la programación de un ordenador en su propio lenguaje. Esto despertó su imaginación: para Peter Samson, un ordenador era sin duda como la lámpara de Aladino: si la frotabas, hacía lo que le pedías. Así que trató de aprender más sobre el tema, construyó sus propias máquinas, participó en concursos y competiciones de proyectos científicos y fue al lugar al que aspiraba la gente de su calaña: el MIT, el depósito de los más brillantes de esos extraños estudiantes de secundaria con gafas de búho y pectorales poco desarrollados que deslumbraban a los profesores de matemáticas y reprobaban Educación Física,

que no soñaban con ponerla en la noche del baile de graduación, sino con llegar a la final del concurso de la Feria de Ciencias de General Electric. En el MIT - donde deambulaba por los pasillos a las dos de la mañana - buscando algo interesante, y donde, en efecto, descubriría algo que le ayudaría a sumergirse profundamente en una nueva forma de proceso creativo y en un nuevo estilo de vida, y que le pondría en la vanguardia de una sociedad imaginada sólo por unos pocos escritores de ciencia ficción de desprestigio cierto. Descubriría un ordenador con el que podría jugar.

La sala EAM con la que Samson se había topado por casualidad estaba llena de grandes máquinas perforadoras de tarjetas del tamaño de archivadores achaparrados. Nadie las protegía: la sala sólo tenía personal durante el día, cuando un grupo selecto que había obtenido la autorización oficial tenía el privilegio de entregar largas tarjetas manila a operadores que luego usarían estas máquinas para perforarlas según los datos que *los privilegiados* quisieran introducir en las tarjetas. Un agujero en la tarjeta representaba alguna instrucción para el ordenador, diciéndole que colocara un dato en algún lugar, o que realizara una función sobre un dato, o que moviera un dato de un lugar a otro. Una pila entera de estas tarjetas formaba un programa de computadora, un programa informático que es una serie de instrucciones que dan lugar a un resultado esperado, al igual que las instrucciones de una receta - seguidas con precisión - dan lugar a un pastel. Esas tarjetas se llevarían a otro operador en el piso de arriba, que introduciría las tarjetas en un "lector" que anotaría dónde estaban los agujeros y enviaría esta información a la IBM 704, en el primer piso del edificio 26. El gigante descomunal.

La IBM 704 costó varios millones de dólares, ocupaba una habitación entera, necesitaba la atención constante de un grupo de operadores de máquinas profesionales y requería un aire acondicionado especial para que los tubos de vacío incandescentes que había en su interior no se calentaran hasta alcanzar temperaturas que destruyeran los datos. Cuando el aire acondicionado se estropeaba (algo bastante habitual), sonaba un fuerte gong y tres ingenieros salían corriendo de una oficina cercana para quitar frenéticamente las tapas de la máquina para que sus entrañas no se derritieran. Todas estas personas encargadas de perforar tarjetas, introducirlas en los lectores y pulsar botones e interruptores de la máquina eran lo que comúnmente se denominaba un *Sacerdocio*, y aquellos con el privilegio de enviar datos a esos sacerdotes tan sagrados eran los *acólitos* oficiales. Era un intercambio casi ritualista.

> ACÓLITO: Oh, máquina, ¿aceptarías mi oferta de información para poder ejecutar mi programa y tal vez darme un cálculo?
>
> SACERDOTE (en nombre de la máquina): Lo intentaremos. No prometemos nada.

Como regla general, ni siquiera a estos acólitos más privilegiados se les permitía el acceso directo a la máquina en sí, y no podían ver durante horas, a veces durante días, los resultados de la ingestión de su "lote" de tarjetas por parte de la máquina.

Esto era algo que Samson sabía, y por supuesto que lo frustraba muchísimo, ya que quería acceder a la maldita máquina. Porque de eso se trataba la vida.

Lo que Samson no sabía - y se alegró de descubrir - era que la sala EAM también contaba con una máquina perforadora particular llamada 407. No sólo podía perforar tarjetas, sino que también podía leerlas, clasificarlas e imprimirlas en listas. Nadie parecía estar vigilando estas máquinas, que eran computadoras, por así decirlo. Por supuesto, usarlas no sería nada fácil: uno necesitaba en realidad

conectar lo que se llamaba un tablero de enchufes, un cuadrado de plástico de cinco por cinco centímetros con una masa de agujeros. Si metes cientos de cables por los agujeros en un orden determinado, obtendrás algo que parece un nido de ratas, pero que encaja en esta máquina electromecánica y modifica su personalidad. Puede hacer lo que quieras que haga.

Así que - sin autorización alguna - eso es lo que Peter Samson se propuso hacer, junto con algunos amigos suyos de una organización del MIT con un interés especial en el modelismo ferroviario. Fue un paso casual e irreflexivo hacia un futuro de ciencia ficción, pero era típico de la forma en que una subcultura extraña se estaba levantando por sus propios medios y creciendo hasta alcanzar una prominencia subterránea, para convertirse en una cultura que sería el alma malcriada y no autorizada del mundo de la informática. Fue una de las primeras aventuras de los hackers del Club de Ferromodelismo, el *Tech Model Railroad Club*, o TMRC.

* * *

Peter Samson había sido miembro del Tech Model Railroad Club desde su primera semana en el MIT en el otoño de 1958. El primer evento al que asistieron los estudiantes de primer año que ingresaban al MIT fue una conferencia de bienvenida tradicional, la misma que se había dado desde que todos en el MIT podían recordar. “MIRA A LA PERSONA A TU IZQUIERDA... MIRA A LA PERSONA A TU DERECHA... UNO DE USTEDES TRES NO SE GRADUARÁ DEL INSTITUTO”. El efecto deseado del discurso era crear esa sensación horrible en la garganta colectiva de los estudiantes de primer año que indicaba un temor sin precedentes. Durante toda su vida, estos estudiantes de primer año habían estado casi exonerados de la presión académica. La exoneración se había ganado en virtud de su brillantez. Ahora, cada uno de ellos tenía una persona a la derecha y una persona a la izquierda que eran igual de inteligentes. Tal vez incluso más inteligentes.

Pero para ciertos estudiantes esto no era un desafío en absoluto. Para estos jóvenes, los compañeros de clase eran percibidos como una especie de neblina amistosa: tal vez pudieran ser de ayuda en la agotadora búsqueda de averiguar cómo funcionaban las cosas y luego dominarlas. Ya había suficientes obstáculos para aprender: ¿para qué molestarse con cosas estúpidas como adular a los profesores y esforzarse por obtener buenas calificaciones? Para estudiantes como Peter Samson, la búsqueda significaba más que el título.

Algún tiempo después de la conferencia llegó la reunión de estudiantes de primer año. Todas las organizaciones del campus (grupos de interés especial, fraternidades y demás) instalaron puestos en un gran gimnasio para tratar de reclutar nuevos miembros. El grupo que atrapó a Peter fue el Tech Model Railroad Club. Sus miembros - estudiantes de último año de ojos brillantes y corte de pelo militar que hablaban con las cadencias espasmódicas de las personas que quieren que las palabras se dejen de lado rápidamente - se jactaban de una exhibición espectacular de trenes de trocha HO que disponían en una sala permanente del club en el Edificio 20. Peter Samson había estado fascinado por los ferrocarriles, especialmente los subterráneos. Así que se unió a la visita guiada a pie hasta el edificio, una estructura temporal revestida de tejas construida durante la Segunda Guerra Mundial. Los pasillos eran cavernosos, y aunque el salón del club estaba en el segundo piso, tenía la sensación húmeda y tenue de un sótano.

El salón del club estaba dominado por una enorme playa de maniobras ferroviaria. Casi llenaba la habitación, y si te parabas en la pequeña área de control llamada "la muesca" podías ver un pequeño pueblo, una pequeña área industrial, una área de trocha de trabajo, una pequeña línea de tranvía en

funcionamiento, una montaña de papel maché y, por supuesto, muchos trenes y vías. Los trenes estaban meticulosamente diseñados para parecerse a sus homólogos de escala real y avanzaban por los recodos de la vía con una perfección digna de un libro de imágenes.

Y entonces Peter Samson miró debajo de los tableros que le llegaban hasta el pecho y que sostenían la playa de maniobras. Se quedó sin aliento. Debajo de esta playa de maniobras estraba la matriz de cables, relés y desvíos más gigantesca de lo que Peter Samson hubiese soñado que existiera. Había líneas de desvíos ordenadas y uniformes, y filas dolorosamente regulares de relés de bronce opaco, y una larga y desordenada maraña de cables rojos, azules y amarillos, que se retorcían y giraban como una explosión de cabello de Einstein con los colores del arco iris. Era un sistema increíblemente complicado y Peter Samson se comprometió a averiguar cómo funcionaba.

Tras registrar cuarenta horas de trabajo en la playa de maniobras a escala, El Tech Model Railroad Club daba a sus miembros una llave de la sala del clube. El estudiante de primer año Midway había estado allí un viernes. Para el lunes, Peter Samson ya tenía su llave.

* * *

Había dos facciones en TMRC. A algunos miembros les encantaba la idea de pasar su tiempo construyendo y pintando réplicas de ciertos trenes con valor histórico y emocional, o creando escenarios realistas para la playa de maniobras. Este era el contingente de la espátula y el pincel, y se suscribía a revistas ferroviarias y reservaban viajes por líneas de trenes dilapidadas a los demás miembros del club. La otra facción se centraba en el Subcomité de Señales y Energía del club, y se preocupaba mucho más por lo que sucedía debajo de la playa de maniobras. Este era *El Sistema*, que funcionaba como una especie de colaboración entre Rube Goldberg y Wernher von Braun, y se mejoraba, renovaba, perfeccionaba y, a veces, se "chocaba" - en la jerga del club, se arruinaba - constantemente. La gente de SyE estaba obsesionada con la forma en que funcionaba El Sistema, sus crecientes complejidades, cómo cualquier cambio que hicieras afectaría a otras partes y cómo podías aprovechar al máximo esas relaciones entre las partes.

Muchas de las piezas para El Sistema habían sido donadas por el Western Electric College Gift Plan, directamente de la compañía telefónica. El asesor de la facultad del club también estaba a cargo del sistema telefónico del campus y se había encargado de que los aficionados al modelismo ferroviario tuvieran a su disposición un equipo telefónico sofisticado. Utilizando ese equipo como punto de partida, los aficionados al modelismo ferroviario habían ideado un esquema que permitía a varias personas controlar los trenes a la vez - incluso si los trenes se encontraban en diferentes partes de la misma vía. Utilizando discos apropiados de teléfonos, los "maquinistas" del TMRC podían especificar qué bloque de vía querían controlar y hacer funcionar un tren desde allí. Esto se hacía utilizando varios tipos de relés de la compañía telefónica, incluidos ejecutores de barras cruzadas e conmutadores paso a paso, que permitían escuchar realmente la energía que se transfería de un bloque a otro mediante un sonido *clac-clac-clac-clac* salido de otro mundo.

Fue el grupo SyE el que ideó este ingenioso plan, y fue el grupo SyE el que albergó el tipo de curiosidad inquieta que los llevó a hurgar en los edificios del campus en busca de formas de hacerse con las computadoras. Fueron discípulos de por vida de un *Imperativo Práctico*. El director de SyE era un estudiante de último año llamado Bob Saunders - de rasgos rubicundos y bulbosos, una risa contagiosa y un talento para los interruptores. De niño en Chicago, había construido un transformador de alta frecuencia para un proyecto de secundaria; era su versión de dos metros de altura de una bobina de Tesla, algo ideado por un ingeniero en el siglo XIX que se suponía que

enviaría furiosas ondas de energía eléctrica. Saunders dijo que su proyecto de bobina logró desintonizar la recepción televisiva en manzanas a la redonda. Otra persona que gravitó hacia SyE fue Alan Kotok - un regordete sin cuello y con gafas gruesas de Nueva Jersey que iba a la clase de Samson. La familia de Kotok lo recordaba - a los tres años - sacando un enchufe de la pared con un destornillador y provocando una sibilante lluvia de chispas. A los seis, construía y conectaba lámparas. Con la escuela secundaria, una vez había visitado el Laboratorio de Investigación Mobil en la cercana Haddonfield, y vio su primer mainframe; la euforia de esa experiencia lo ayudó a decidir entrar en el MIT. En su primer año, se ganó la reputación de ser uno de los especialistas en SyE más capaces del TMRC.

Los especialistas en SyE eran los que pasaban los sábados yendo al depósito de chatarra de Eli Heffron en Somerville desenterrando piezas, los que pasaban horas tumbados boca arriba sobre pequeñas sillas con ruedas a las que llamaban "literas" para meterse debajo de los puntos estrechos del sistema de conmutación, los que trabajaban durante la noche haciendo la conexión absolutamente desautorizada entre el teléfono del TMRC y el Campus Este. La tecnología era su patio de recreo.

Los miembros principales pasaban horas en el club, mejorando constantemente El Sistema, discutiendo sobre lo que se podría hacer a continuación, desarrollando una jerga propia que parecía incomprensible para los forasteros que pudieran toparse con estos fanáticos adolescentes, con sus camisas de manga corta a cuadros, lápices en sus bolsillos, pantalones chinos y, siempre, con una botella de Coca-Cola a su lado. (El TMRC compró su propia máquina de Coca-Cola por la entonces prohibitiva suma de 165 dólares - a una tarifa de cinco centavos por botella, el gasto se reponía en tres meses - para facilitar las ventas, Saunders construyó una máquina de cambio para los compradores de Coca-Cola que una década después todavía estaba en uso). Cuando un equipo no funcionaba, estaba "perdiendo"; cuando un equipo se estropeaba, se "Hacía Puré" (Hacerlo Puré hasta que no Sirva); los dos escritorios en la esquina de la habitación no se llamaban la oficina, sino el "orificio"; aquel que insistía en estudiar para los cursos era una "herramienta"; la basura se llamaba "costra"; y un proyecto emprendido o un producto construido no solo para cumplir algún objetivo constructivo, sino con algún placer salvaje obtenido por la mera participación, se llamaba "hack".

Este último término puede haber sido sugerido por la antigua jerga del MIT: la palabra "hack" (“hachazo”) se había utilizado durante mucho tiempo para describir las elaboradas travesuras universitarias que los estudiantes del MIT solían idear, como cubrir la cúpula que daba al campus con papel aluminio. Pero tal como la gente del TMRC usaba la palabra, había un gran respeto implícito. Si bien alguien podría llamar a una conexión inteligente entre relés un "simple hack", se entendería que - para calificar como *hack* - la hazaña debía estar imbuida de innovación, estilo y virtuosismo técnico. Aunque uno pudiera decir con autodesprecio que estaba "hackeando El Sistema" (de la misma manera que un hacha corta troncos), se reconocía que el arte con el que uno *hackeaba* era considerable.

Las personas más productivas que trabajaban en Señales y Energía se llamaban a sí mismas "hackers" con gran orgullo. En los confines del salón del club en el Edificio 20, y de la "Sala de Herramientas" (donde se llevaban a cabo algunos estudios y muchas sesiones de armado tecnológico), se habían dotado unilateralmente de los atributos heroicos de la leyenda islandesa. Así es como Peter Samson se veía a sí mismo y a sus amigos en un poema al estilo Sandburg en el boletín del club:

> Lanzador de Interruptores para el Mundo,
> Probador de Fusibles, Creador de Rutas,
> Jugador con los Ferrocarriles y el Chopper Avanzados del Sistema;
> Sucio, peludo, desparramado,
> Máquina de la Función de Punto, Línea de Luz:
> Me dicen que eres malvado y les creo; porque he visto tus bombillas pintadas bajo la metacrilato atrayendo a los coolies del sistema...
> Bajo la torre, polvo por todas partes, hachando con resortes bifurcados...
> Hackeando como actúa un novato ignorante que nunca ha perdido la ocupación y ha abandonado
> Hackeando los Tableros M, pues bajo sus cerraduras están los interruptores, y bajo su control el avance en la playa de maniobras,
> ¡Hackeando!
> Hackeando los sucios, peludos y desgarbados hacks de la juventud; friendo diodos descableados, orgullosos de ser el Lanza-cambios, el Probador de Fusibles, el Creador de Rutas, el Jugador con Ferrocarriles y el Chopper Avanzado del Sistema.

Siempre que podían, Samson y los demás se escabullían a la sala EAM con sus tableros de enchufes, tratando de usar la máquina para llevar un registro de los interruptores debajo de la playa de maniobras. Igualmente importante, estaban viendo lo que podía hacer el contador electromecánico, llevándolo al límite.

Esa primavera de 1959, se ofreció un nuevo curso en el MIT. Era el primer curso de programación de una computadora que podían tomar los estudiantes de primer año. El profesor era un hombre distante con una mata de pelo salvaje y una barba igualmente rebelde: John McCarthy. McCarthy era un profesor de matemáticas magistral, típicamente distraído; abundaban las historias sobre su hábito de responder de repente a una pregunta horas, a veces incluso días después de que se la hubieran planteado por primera vez. Se acercaba a uno en el pasillo y, sin saludar, empezaba a hablar con su dicción robóticamente precisa, como si la pausa en la conversación hubiera sido sólo una fracción de segundo y no una semana. Lo más probable es que su respuesta tardía fuera brillante.

McCarthy era una de las pocas personas que trabajaban en una forma completamente nueva de investigación científica con ordenadores. La naturaleza volátil y controvertida de su campo de estudio era obvia por la arrogancia del nombre que McCarthy le había otorgado: Inteligencia Artificial. Este hombre realmente pensaba que los ordenadores podían ser *listas*. Incluso en un lugar con tanta intensidad científica como el MIT, la mayoría de la gente consideraba ridícula la idea: consideraban que las computadoras eran herramientas útiles, aunque un tanto absurdamente caras, para realizar cálculos enormes y para diseñar sistemas de defensa antimisiles (como el ordenador más grande del MIT, el *Whirlwind*, había hecho para el sistema de alerta temprana SAGE), pero se burlaban de la idea de que las computadoras en sí mismas pudieran ser en realidad un campo de estudio científico. La informática no existía oficialmente en el MIT a finales de los años cincuenta, y McCarthy y sus colegas especialistas en informática trabajaban en el Departamento de Ingeniería Eléctrica, que ofrecía el curso, el número 641, que Kotok, Samson y algunos otros miembros del TRMC tomaron esa primavera.

McCarthy había iniciado un programa gigantesco sobre la IBM 704, el enorme gigante, que le daría la extraordinaria capacidad de jugar al ajedrez. Para los críticos del campo en ciernes de la

inteligencia artificial, este era sólo un ejemplo del optimismo estúpido de gente como John McCarthy. Pero McCarthy tenía una cierta visión de lo que podían hacer las computadoras, y jugar al ajedrez era sólo el comienzo.

Todo era fascinante, pero no la visión que impulsaba a Kotok, Samson y los demás. Querían aprender a *hacer funcionar* las malditas máquinas, y aunque este nuevo lenguaje de programación llamado LISP del que hablaba McCarthy en el 641 era interesante, no era ni de lejos tan interesante como el acto de programar, o ese momento fantástico en el que recibías la copia impresa del Sacerdocio (¡palabras de la propia fuente!) y podías pasar horas estudiando los resultados del programa, qué había ido mal, cómo se podía mejorar. Los hackers del TMRC ideaban formas de entrar en contacto más estrecho con la IBM 704, que pronto fue actualizada a un modelo más nuevo llamado 709. Pasando el rato en el centro de cómputo a altas horas de la madrugada, conociendo al Sacerdocio y haciendo reverencias y raspando el suelo el número de veces necesario, a gente como Kotok se le permitió finalmente pulsar algunos botones de la máquina y observar cómo funcionaban las luces.

Había secretos sobre esas máquinas IBM que habían sido aprendidos con mucho esfuerzo por algunas de las personas mayores del MIT con acceso al 704 y amigos entre el Sacerdocio. Sorprendentemente, algunos de estos programadores, estudiantes de posgrado que trabajaban con McCarthy, incluso habían escrito un programa que utilizaba una de las filas de luces diminutas: las luces se encendían en tal orden que parecía como si una pequeña pelota estuviera pasando de derecha a izquierda: si un operador pulsaba un interruptor en el momento justo, el movimiento de las luces podía invertirse: ¡ping-pong informático! Obviamente, este era el tipo de cosas que uno mostraba para impresionar a sus compañeros, quienes luego echaban un vistazo al programa real que había escrito y veían cómo se hacía.

Para mejorar el programa, alguien más podía intentar hacer lo mismo con menos instrucciones, un esfuerzo que valía la pena, ya que había tan poco espacio en la pequeña "memoria" de las computadoras de aquellos días que no cabían muchas instrucciones en ellas. John McCarthy había notado una vez cómo sus estudiantes de posgrado que deambulaban por el 704 trabajaban en sus programas de computadora para aprovechar al máximo la menor cantidad de instrucciones y comprimir el programa de modo que fuera necesario introducir menos tarjetas en la máquina. Eliminar una instrucción o dos era casi una obsesión para ellos. McCarthy comparó a estos estudiantes con los esquiadores; obtenían el mismo tipo de emoción primaria al "maximizar el código" que los esquiadores fanáticos al deslizarse frenéticamente por una colina. Así, la práctica de tomar un programa de ordenador y tratar de cortar instrucciones sin afectar al resultado pasó a llamarse "programación abreviado", y a menudo se oía a gente murmurar cosas como "Quizá pueda programar unas cuantas instrucciones y conseguir que el cargador de tarjetas de corrección octal reduzca a tres tarjetas en lugar de cuatro".

En 1959, McCarthy estaba cambiando su interés del ajedrez hacia una nueva forma de comunicarse con el ordenador, el nuevo "lenguaje" llamado *LISP*. Alan Kotok y sus amigos estaban más que ansiosos por hacerse cargo del proyecto de ajedrez. Trabajando en el IBM de procesamiento por lotes, se embarcaron en el gigantesco proyecto de enseñar al 704 - y más tarde al 709 - e incluso después de éste a su sustituto, el 7090, cómo jugar al juego de los reyes. Con el tiempo, el grupo de Kotok se convirtió en el mayor usuario de tiempo de cómputo de todo el centro de cómputo del MIT.

Aun así, trabajar con la máquina IBM era frustrante. No había nada peor que la larga espera entre el momento en que entregabas tus cartas y el momento en que te devolvían los resultados. Si se hubiera escrito mal una sola letra en una instrucción, el programa se bloquearía y habría que empezar todo el proceso de nuevo. Esto iba de la mano con la asfixiante proliferación de malditas *reglas* que impregnaban la atmósfera del centro de computación. La mayoría de las reglas estaban diseñadas para mantener a los jóvenes fanáticos de las computadoras como Samson, Kotok y Saunders físicamente alejados de la máquina en sí. La regla más rígida de todas era que nadie debería poder tocar o manipular la máquina en sí. Esto, por supuesto, era lo que la gente de Señales y Energía se moría por hacer más que cualquier otra cosa en el mundo, y las restricciones los volvían locos.

Un sacerdote (un suboficial de bajo nivel, en realidad) en el turno de noche era particularmente desagradable al hacer cumplir esta regla, por lo que Samson ideó una venganza adecuada. Mientras husmeaba en la tienda de chatarra electrónica de Eli un día, se topó con un tablero eléctrico exactamente como el tipo de tablero que contenía los toscos tubos de vacío que se encontraban dentro de la IBM. Una noche, antes de las 4 a. m., este subsacerdote en particular salió por un minuto; cuando regresó, Samson le dijo que la máquina no estaba funcionando, pero que habían encontrado el problema y levantó el módulo totalmente destrozado del viejo 704 que había conseguido en Eli's.

El subsacerdote apenas podía pronunciar las palabras. "¿D-dónde conseguiste eso?"

Samson, que tenía grandes ojos verdes que fácilmente podían parecer maníacos, señaló lentamente un lugar abierto en la máquina donde – por supuesto – nunca había habido una placa, pero el lugar aún parecía tristemente desnudo.

El suboficial se quedó boquiabierto, hizo muecas que indicaban que sus entrañas estaban a punto de rendirse y gimió exhortando a la deidad. Sin duda, visiones de una deducción de un millón de dólares de su sueldo comenzaron a desfilar ante él. Sólo después de que su supervisor, un *Sumo Sacerdote* con cierta comprensión de la mentalidad de estos jóvenes sabelotodos del Club del Ferrocarril Modelo, vino y le explicó la situación, se calmó.

No sería el último administrador en sentir la ira de un hacker frustrado su intento de acceso.

* * *

Un día, un ex miembro del TMRC que ahora formaba parte del cuerpo docente del MIT visitó la sala del club. Su nombre era Jack Dennis. Cuando era estudiante universitario a principios de los años 50, había trabajado frenéticamente debajo del diseño. Dennis últimamente había estado trabajando con una computadora que el MIT acababa de recibir del Lincoln Lab, un laboratorio de desarrollo militar afiliado al Instituto. El ordenador se llamaba TX-0 y fue uno de los primeros ordenadores del mundo que funcionaban con transistores. El Laboratorio Lincoln lo había utilizado específicamente para probar un ordenador gigante llamado TX-2, que tenía una memoria tan compleja que sólo con este hermano pequeño especialmente construido se podían diagnosticar sus problemas. Ahora que su trabajo original había terminado, el TX-0, que costó tres millones de dólares, había sido enviado al Instituto en "préstamo a largo plazo" y, aparentemente, nadie en el Laboratorio Lincoln había marcado un calendario con una fecha de devolución. Dennis preguntó a la gente de SyE en TMRC si les gustaría verlo.

¡Eh, monjas! ¿Les gustaría conocer al Papa?

La TX-0 estaba en el Edificio 26, en el Laboratorio de Radio Electrónica (RLE) del segundo piso, directamente encima del Centro de Cálculo del primer piso, que albergaba el enorme IBM 704. El laboratorio RLE se parecía a la sala de control de una nave espacial antigua. El TX-0, o Tixo, como se lo llamaba a veces, era para su época una máquina enana, ya que fue uno de los primeros ordenadores que utilizaba transistores del tamaño de un dedo en lugar de tubos de vacío del tamaño de una mano. Aun así, ocupaba gran parte de la habitación, junto con sus quince toneladas de equipo de aire acondicionado de apoyo. Los mecanismos del TX-O estaban montados sobre varios chasis altos y delgados, como estanterías de metal resistentes, con cables enredados y pequeñas hileras ordenadas de diminutos contenedores con forma de botella en los que se insertaban los transistores. Otro estante tenía un frente de metal sólido salpicado de indicadores de aspecto lúgubre. Frente a los estantes había una consola en forma de L, el panel de control de esta nave espacial de H. G. Wells, con un mostrador azul para los codos y los papeles. En el brazo corto de la L había una Flexowriter, que parecía una máquina de escribir adaptada para la guerra de tanques, con su parte inferior anclada en una carcasa gris militar. Sobre la parte superior estaban los paneles de control, protuberancias con forma de caja pintadas de un amarillo institucional. En los laterales de las cajas que se encontraban frente al usuario había unos cuantos indicadores, varias líneas de luces parpadeantes de un cuarto de pulgada, una matriz de interruptores de palanca de acero del tamaño de granos de arroz grandes y, lo mejor de todo, una pantalla de tubo de rayos catódicos real, redonda y de color gris humo.

La gente de TMRC estaba asombrada. *Está máquina no utilizaba tarjetas*. El usuario primero introducía un programa en una cinta de papel larga y delgada con una Flexowriter (había algunas Flexowriters adicionales en una habitación contigua), luego se sentaba en la consola, introducía el programa haciendo pasar la cinta por un lector y podía sentarse allí mientras se ejecutaba el programa. Si algo salía mal con el programa, se sabía de inmediato y se podía diagnosticar el problema utilizando algunos de los interruptores o comprobando cuáles de las luces parpadeaban o estaban encendidas. La computadora incluso tenía una salida de audio: mientras se ejecutaba el programa, un altavoz debajo de la consola emitía una especie de música, como un órgano eléctrico mal afinado cuyas notas vibraban con un estruendo difuso y etéreo. Los acordes de este "órgano" cambiaban según los datos que la máquina estuviera leyendo en un microsegundo determinado; una vez que te familiarizabas con los tonos, podías *oír* en realidad en qué parte del programa estaba trabajando la computadora. Sin embargo, tenías que discernir esto por encima del traqueteo de la Flexowriter, que podía hacerte pensar que estabas en medio de una batalla de ametralladoras.

Aún más sorprendente era que, debido a estas capacidades "interactivas", y también porque los usuarios parecían tener bloques de tiempo para usar el TX-0 por sí solos, incluso podías modificar un programa *mientras estás sentado ante la computadora*. ¡Un milagro!

No había forma de que Kotok, Saunders, Samson y los demás se mantuvieran alejados de esa máquina. Afortunadamente, no parecía haber el tipo de burocracia alrededor del TX-0 que había alrededor del IBM 704. Ningún grupo de sacerdotes oficiosos. El técnico a cargo era un astuto escocés de pelo blanco llamado John McKenzie. Si bien se aseguró de que los estudiantes de posgrado y aquellos que trabajaban en proyectos financiados (Usuarios Oficialmente Sancionados) mantuvieran el acceso a la máquina, McKenzie toleró al grupo de locos de TMRC que comenzaron a frecuentar el laboratorio RLE, donde se encontraba en pie el TX-0.

Samson, Kotok, Saunders y un estudiante de primer año llamado Bob Wagner pronto se dieron cuenta de que el mejor momento para pasar el rato en el Edificio 26 era por la noche, cuando

ninguna persona en su sano juicio se habría apuntado a una sesión de una hora en el papel que se colocaba todos los viernes junto al aire acondicionado en el laboratorio de RLE. El TX-0, por regla general, se mantenía en funcionamiento veinticuatro horas al día: las computadoras de entonces eran demasiado caras para perder el tiempo dejándolas inactivas durante la noche y, además, era un procedimiento complicado ponerlas en funcionamiento una vez apagadas. Así que los hackers de TMRC, que pronto se referían a sí mismos como hackers TX-0, cambiaron su estilo de vida para adaptarse al ordenador. Se apropiaban de los bloques de tiempo que podían y "aprovechaban el tiempo" con visitas nocturnas al laboratorio por si acaso alguien que tenía programada una sesión a las 3 de la mañana no aparecía.

"¡Oh!" Samson decía encantado, un minuto o dos después de que alguien no se presentase a la hora designada en el libro de registro. "¡Asegúrate de que no se desperdicie!"

Nunca parecía que se desperdiciara, porque los hackers estaban allí casi todo el tiempo. Si no estaban en el laboratorio de RLE esperando a que se produjera una vacante, estaban en el aula contigua a la sala del club de TMRC, la Sala de Herramientas, jugando a un juego de palabras al estilo del "ahorcado" que Samson había ideado llamado "Ven a la puerta de al lado", esperando una llamada de alguien que estuviera cerca del TX-0, vigilándolo para ver si alguien no se había presentado a una sesión. Los hackers reclutaban una red de informantes para dar aviso previo de posibles vacantes en el ordenador: si un proyecto de investigación no estaba listo con su programa a tiempo, o un profesor estaba enfermo, se pasaba la voz a TMRC y los hackers aparecían en el TX-0, sin aliento y listos para meterse en el espacio detrás de la consola.

Aunque Jack Dennis estaba teóricamente a cargo de la operación, Dennis estaba dando cursos en ese momento y prefería pasar el resto de su tiempo escribiendo códigos para la máquina. Dennis jugó el papel de padrino benévolo de los hackers: les daba una breve introducción práctica a la máquina, les indicaba ciertas direcciones, se divertía con sus locas aventuras de programación. Sin embargo, no tenía gusto por la administración y estaba igualmente feliz de dejar que John McKenzie manejara las cosas. McKenzie reconoció pronto que la naturaleza interactiva del TX-0 estaba inspirando una nueva forma de programación informática y los hackers eran sus pioneros. Por eso no dictó demasiados edictos.

El ambiente era lo suficientemente relajado en 1959 como para dar cabida a los extraviados: personas locas por la ciencia cuya curiosidad ardía como el hambre, que, como Peter Samson, estarían explorando el laberinto inexplorado de laboratorios del MIT. El ruido del aire acondicionado, la salida de audio y el taladro-martillo Flexowriter atraían a estos vagabundos, que metían la cabeza en el laboratorio como gatitos que miran dentro de cestas de lana.

Uno de esos vagabundos era un forastero llamado Peter Deutsch. Incluso antes de descubrir el TX-0, Deutsch había desarrollado una fascinación por las computadoras. Comenzó un día cuando cogió un manual que alguien había descartado, un manual para una forma oscura de lenguaje informático para hacer cálculos. Algo en el orden de las instrucciones de la computadora le atraía: más tarde describiría la sensación como el mismo tipo de reconocimiento inquietantemente trascendente que experimenta un artista cuando descubre el medio que es absolutamente adecuado para él. ESTE ES EL LUGAR AL QUE PERTENEZCO. Deutsch intentó escribir un pequeño programa y, firmando una cita con el nombre de uno de los sacerdotes, lo ejecutó en una computadora. En cuestión de semanas, había alcanzado una sorprendente habilidad en programación. Tenía solo doce años.

Era un chico tímido, bueno en matemáticas e inseguro en casi todo lo demás. Tenía un sobrepeso incómodamente gordo, era deficiente en deportes, pero era un gran intelectual. Su padre era profesor en el MIT, y Peter usó eso como entrada para explorar los laboratorios.

Era inevitable que se sintiera atraído por el TX-0. Primero se dirigió a la pequeña "sala Kluge" (un "kluge" es una pieza de equipo construida de manera poco elegante que parece desafiar la lógica al funcionar correctamente), donde había tres Flexowriters fuera de línea disponibles para perforar programas en cinta de papel que luego se introducirían en el TX-0. Alguien estaba ocupado perforando una cinta. Peter observó durante un rato, luego comenzó a bombardear al pobre alma con preguntas sobre esa pequeña computadora de aspecto extraño en la habitación de al lado. Luego Peter se acercó al TX-0, lo examinó de cerca y notó en qué se diferenciaba de otras computadoras: era más pequeño, tenía una pantalla CRT y otros juguetes interesantes. Decidió en ese momento actuar como si tuviera todo el derecho a estar allí. Consiguió un manual y pronto sorprendió a la gente con sus palabras sobre informática que tenían sentido y, finalmente, le permitieron inscribirse en sesiones nocturnas y de fin de semana y escribir sus propios programas.

A McKenzie le preocupaba que alguien pudiera acusarlo de organizar una especie de campamento de verano con este pequeño de pantalones cortos, apenas lo suficientemente alto como para sacar la cabeza por encima de la consola del TX-0, miraba el código que un Usuario Oficialmente Sancionado, tal vez algún estudiante de posgrado engreído, estaba martillando en la Flexowriter, y decía con su voz chillona de preadolescente algo como "Tu problema es que este crédito está mal aquí... necesitas esta otra instrucción allí", y el estudiante de posgrado engreído se volvía loco - ¿QUIÉN ES ESTE PEQUEÑO GUSANO?- y comenzaba a gritarle que saliera a jugar a algún lado. Sin embargo, invariablemente los comentarios de Peter Deutsch resultaban ser correctos. Deutsch también anunciaba descaradamente que iba a escribir mejores programas que los que estaban disponibles en ese momento, y él iba y lo hacía.

Samson, Kotok y los otros hackers aceptaron a Peter Deutsch: en virtud de su conocimiento de computadoras, era digno de un trato igualitario. Deutsch no era tan querido por los usuarios oficialmente autorizados, especialmente cuando se sentaba detrás de ellos, listo para entrar en acción cuando cometían un error en la Flexowriter.

Estos usuarios oficialmente autorizados aparecían en el TX-0 con la regularidad de los viajeros. Los programas que ejecutaban eran análisis estadísticos, correlaciones cruzadas, simulaciones del interior del núcleo de una célula. Aplicaciones. Eso estaba bien para los usuarios, pero era una especie de desperdicio en las mentes de los hackers. Lo que los hackers tenían en mente era ponerse detrás de la consola del TX-0 de la misma manera que se pone detrás del acelerador de un avión. O, como dijo Peter Samson, un aficionado a la música clásica, calcular con el TX-0 era como tocar un instrumento musical: un instrumento musical absurdamente caro con el que se podía improvisar, componer y, como los beatniks de Harvard Square a una milla de distancia, gemir como un alma en pena con total abandono creativo.

Una cosa que les permitió hacer esto fue el sistema de programación ideado por Jack Dennis y otro profesor, Tom Stockman. Cuando el TX-0 llegó al MIT, había sido desmantelado desde sus días en el Lincoln Lab: la memoria se había reducido considerablemente, a 4.096 "palabras" de dieciocho bits cada una. (Un "bit" es un dígito binario, ya sea un uno o un cero. Estos números binarios son lo único que entienden las computadoras. Una serie de números binarios se llama "palabra"). Y el TX-0 casi no tenía software. Por lo tanto, Jack Dennis, incluso antes de presentar el TX-0 a la gente del

TMRC, había estado escribiendo "programas de sistemas", el software para ayudar a los usuarios a utilizar la máquina.

Lo primero en lo que trabajó Dennis fue en un ensamblador. Esto era algo que traducía el *lenguaje ensamblador* (que utilizaba abreviaturas simbólicas de tres letras que representaban instrucciones para la máquina) al lenguaje de máquina, que consistía en los números binarios 0 y 1. El TX-0 tenía un lenguaje ensamblador bastante limitado: dado que su diseño permitía utilizar sólo dos bits de cada palabra de dieciocho bits para las instrucciones al ordenador, sólo se podían utilizar cuatro instrucciones (cada posible variación de dos bits (00, 0, 1, 10 y 11) representaba una instrucción). Todo lo que hacía el ordenador podía descomponerse en la ejecución de una de esas cuatro instrucciones: hacía falta una instrucción para sumar dos números, pero una serie de quizá veinte instrucciones para multiplicar dos números. Mirar fijamente una larga lista de comandos de ordenador escritos como números binarios (por ejemplo, 10011001100001) podía convertirte en un caso mental balbuceante en cuestión de minutos. Pero el mismo comando en lenguaje ensamblador podría verse así: ADD Y. Después de cargar la computadora con el ensamblador que Dennis escribió, usted podría escribir programas en esta forma simbólica más simple y esperar con satisfacción mientras la computadora hacía la traducción al binario por usted. Luego, usted alimentaría ese código "objeto" binario nuevamente a la computadora. El valor de esto era incalculable: permitía a los programadores escribir algo que PARECÍA código, en lugar de una serie interminable y vertiginosa de 1s y 0s.

El otro programa en el que Dennis trabajó con Stockman era algo aún más nuevo: un depurador. El TX-0 venía con un programa de depuración llamado UT-3, que le permitía hablar con la computadora mientras estaba funcionando escribiendo comandos directamente en la Flexowriter, pero tenía problemas terribles: por un lado, solo aceptaba código escrito que usara el sistema numérico octal. "Octal" es un sistema numérico de base ocho (a diferencia del binario, que es de base dos, y del árabe, el nuestro, que es de base diez), y es un sistema difícil de usar. Por eso Dennis y Stockman decidieron escribir algo mejor que UT-3 que permitiera a los usuarios usar el lenguaje ensamblador simbólico, más fácil de usar. Esto se llamó FLIT y permitía a los usuarios encontrar errores de programa durante una sesión, corregirlos y mantener el programa en funcionamiento. (Dennis explicaría que "FLIT" significaba FLexowriter Interrogation Tape, pero claramente el origen real del nombre era el spray para insectos con esa marca). FLIT fue un gran salto adelante, ya que liberó a los programadores para que realmente hicieran composiciones originales en la máquina, al igual que los músicos componen en sus instrumentos musicales. Gracias al depurador, que ocupaba un tercio de las 4.096 palabras de la memoria del TX-0, los usuarios podían crear sus propias composiciones en la máquina, al igual que los músicos componen en sus instrumentos musicales. Los hackers tenían libertad para crear un nuevo y más atrevido estilo de programación.

¿Y qué hacían estos programas de hackers? Bueno, a veces no importaba demasiado lo que hacían. Peter Samson se pasó la noche hackeando un programa que convertía instantáneamente números arábigos en romanos, y Jack Dennis, después de admirar la habilidad con la que Samson había logrado esta hazaña, dijo: "Dios mío, ¿por qué alguien querría hacer algo así?" Pero Dennis sabía por qué. Había amplia justificación en la sensación de poder y logro que Samson obtuvo cuando introdujo la cinta de papel, controló las luces y los interruptores, y vio que lo que antes eran simples números arábigos de pizarra volvían a ser los números con los que los romanos habían hackeado.

De hecho, fue Jack Dennis quien le sugirió a Samson que existían usos considerables para la capacidad del TX-0 de enviar ruido al altavoz de audio. Aunque no había controles integrados para

el tono, la amplitud o el carácter del sonido, había una forma de controlar el altavoz: se emitían sonidos dependiendo del estado del decimocuarto bit de las palabras de dieciocho bits que el TX-0 tenía en su acumulador en un microsegundo determinado. El sonido se activaba o desactivaba dependiendo de si el bit catorce era un uno o un cero. Así que Samson se puso a escribir programas que variaban los números binarios en esa ranura de diferentes maneras para producir diferentes tonos.

En ese momento, solo unas pocas personas en el país habían estado experimentando con el uso de una computadora para producir cualquier tipo de música, y los métodos que habían estado usando requerían cálculos masivos antes de que la máquina pudiera siquiera emitir una nota. Samson, que reaccionó con impaciencia a quienes le advirtieron que estaba intentando lo imposible, quería una computadora que reprodujera música de inmediato. Así que aprendió a controlar ese bit en el acumulador con tanta destreza que podía comandarlo con la autoridad de Charlie Parker en el saxofón. En una versión posterior de este compilador de música, Samson lo modificó de modo que si uno cometía un error en la sintaxis de programación, la Flexowriter cambiara a una cinta roja e imprimiera "Errar es humano, perdonar es divino".

Cuando los extraños escuchaban las melodías de Johann Sebastian Bach en una onda cuadrada monofónica de una sola voz, sin armonía, no se inmutaban. ¡Vaya cosa! Tres millones de dólares por este gigantesco trozo de maquinaria, ¿y por qué no iba a hacer al menos lo mismo que un piano de juguete de cinco dólares? No tenía sentido explicarles a esos extraños que Peter Samson prácticamente había pasado por alto el proceso mediante el cual se había hecho música durante eones. La música siempre se había hecho creando directamente vibraciones que eran sonido. Lo que sucedía en el programa de Samson era que un montón de números, bits de información que se introducían en una computadora, formaban un código en el que residía la música. Uno podía pasar horas mirando el código y no ser capaz de adivinar dónde estaba la música. Sólo se convirtió en música mientras millones de intercambios de datos brevísimos se llevaban a cabo en el acumulador ubicado en uno de los bastidores de metal, alambre y silicio que componían el TX-0. Samson le había pedido al ordenador, que no tenía ningún conocimiento aparente de cómo utilizar una voz, que se elevara en una canción, y el TX-0 había obedecido.

De modo que un programa de ordenador no sólo era metafóricamente una composición musical, ¡era LITERALMENTE una composición musical! Parecía (y era) el mismo tipo de programa que producía cálculos aritméticos complejos y análisis estadísticos. Estos dígitos que Samson había introducido en el ordenador eran un lenguaje universal que podía producir CUALQUIER COSA: una fuga de Bach o un sistema antiaéreo.

Samson no dijo nada de esto a los extraños que no se impresionaron con su hazaña. Ni siquiera los hackers mismos lo comentaron; ni siquiera está claro que analizaran el fenómeno en términos tan cósmicos. Peter Samson lo hizo, y sus colegas lo apreciaron, porque obviamente fue un hackeo ingenioso. Eso fue justificación suficiente.

* * *

Para hackers como Bob Saunders —discípulo calvo, regordete y alegre, presidente del grupo SyE de TMRC, estudiante de sistemas— la presencia del TX-0 representaba una existencia perfecta. Saunders había crecido en los suburbios de Chicago y —desde que tenía memoria— el funcionamiento de la electricidad y los circuitos telefónicos le habían fascinado. Antes de empezar

en el MIT, Saunders había conseguido un trabajo de verano de ensueño, trabajando para la compañía telefónica instalando equipos de la oficina central. Pasaba ocho horas maravillosas con un soldador y unos alicates en la mano, trabajando en las entrañas de varios sistemas, un idilio interrumpido por las horas de almuerzo dedicadas al estudio profundo de los manuales de la compañía telefónica. Fue el equipamiento de la compañía telefónica debajo de la playa de maniobras del TMRC el que convenció a Saunders de participar activamente en el Club de Ferrocarriles Modelo.

Saunders, siendo un estudiante de último año, había llegado al TX-0 más tarde en su carrera universitaria que Kotok y Samson: había usado el respiro para sentar las bases de una vida social — que incluía el noviazgo y eventual matrimonio con Marge French, quien había hecho algunos trabajos informáticos que no eran hacking para un proyecto de investigación. Aun así, el TX-0 era el centro de su carrera universitaria, y compartía la experiencia común de los hackers de ver cómo sus notas se veían afectadas por las clases que no asistía. No le molestaba mucho, porque sabía que su verdadera educación se llevaba a cabo en la Sala 240 del Edificio 26, detrás de la consola del TX-0. Años más tarde se describiría a sí mismo y a los demás como "un grupo de élite. Otras personas estaban estudiando, pasando sus días en edificios de cuatro pisos produciendo vapores desagradables o en el laboratorio de física arrojando partículas a cosas o lo que sea que fuesen. Y simplemente no prestábamos atención a lo que otras personas estaban haciendo porque no nos interesaba. Ellos estudiaban lo que estaban estudiando y nosotros estudiábamos lo que estábamos estudiando. Y el hecho de que gran parte de eso no estuviera en el plan de estudios aprobado oficialmente era en gran medida irrelevante".

Los hackers salían por la noche. Era la única manera de aprovechar al máximo las cruciales "horas libres" del TX-0. Durante el día, Saunders solía hacer acto de presencia en una o dos clases. Luego dedicaba un tiempo a realizar "mantenimiento básico", como comer e ir al baño. Podía ver a Marge durante un rato, pero finalmente se dirigía al edificio 26. Revisaba algunos de los programas de la noche anterior, impresos en el papel de 24,13 centímetros de ancho que utilizaba la Flexowriter. Anotaba y modificaba la lista para actualizar el código a lo que él consideraba la siguiente etapa de funcionamiento. Tal vez se trasladaba luego al Club de Ferrocarriles en Miniatura e intercambiaba su programa con alguien, comprobando simultáneamente las buenas ideas y los posibles errores. Luego volvía al edificio 26, a la sala Kluge junto al TX-0, para encontrar una Flexowriter fuera de línea en la que actualizar su código. Durante todo el tiempo, comprobaba si alguien había cancelado una sesión de una hora en la máquina; su propia sesión estaba programada para las dos o las tres de la mañana. Esperaba en la sala Kluge o jugaba al bridge en el Railroad Club hasta que llegase el momento.

Sentado en la consola, frente a los estantes de metal que contenían los transistores del ordenador, cada transistor representando una ubicación que contenía o no un bit de memoria, Saunders configuraba la Flexowriter, que lo saludaba con la palabra "WALRUS". Esto era algo que Samson había hackeado en honor al poema de Lewis Carroll con el verso "Ha llegado el momento, dijo la morsa...". Saunders podría reírse de eso mientras buscaba en el cajón la cinta de papel perforado que contenía el programa ensamblador y la introducía en el lector de cintas. Entonces el ordenador estaría listo para ensamblar su programa, así que tomaba la cinta Flexowriter en la que había estado trabajando y la enviaba al ordenador. Observaba cómo se encendían las luces mientras el ordenador cambiaba su código de "fuente" (el lenguaje ensamblador simbólico) a código "objeto" (binario), el

cual el ordenador perforaba en otra cinta de papel. Como esa cinta estaba en el código objeto que entendía el TX-0, la introduciría, con la esperanza de que el programa se ejecutara magníficamente.

Lo más probable es que hubiera algunos hackers detrás de él, riendo y bromeando y bebiendo Coca-Cola y comiendo comida chatarra que habían extraído de la máquina de abajo. Saunders prefería las rodajas de gelatina de limón que los demás llamaban "lemon gunkies". Pero a las cuatro de la mañana, cualquier cosa sabía bien. Todos observaban cómo el programa comenzaba a ejecutarse, las luces se encendían, el altavoz zumbaba en un registro alto o bajo dependiendo de lo que hubiera en el bit 14 del acumulador, y lo primero que veía en la pantalla CRT una vez que el programa se ensamblaba y ejecutaba, era que el programa se había bloqueado. Entonces buscaba la cinta con el depurador FLIT en el cajón y la introducía en la computadora. El mainframe se convertiría entonces en una máquina de depuración y él le volvía a introducir el programa nuevamente. Entonces podía empezar a intentar averiguar en qué lugar habían ido mal las cosas y - si tenía suerte - quizá lo averiguaba y modificaba las cosas introduciendo algunos comandos, pulsando algunos de los interruptores de la consola en un orden preciso, o dando entrada a cierto código en la Flexowriter. Una vez que todo se ponía en marcha (y siempre resultaba increíblemente satisfactorio cuando algo funcionaba, cuando había conseguido que esa sala llena de transistores, cables, metal y electricidad se fusionaran para crear un resultado preciso que él había ideado), intentaba añadirle el siguiente avance. Cuando terminaba la hora (con alguien ya ansiando subirse a la máquina después de él), Saunders estaría listo para pasar las siguientes horas averiguando qué diablos había hecho que el programa se fuera a pique.

La hora pico en sí misma era tremendamente intensa, pero durante las horas anteriores e incluso durante las horas posteriores, un hacker alcanzaba un estado de *concentración pura*. Cuando se programaba una computadora, había que saber a dónde iban todos los miles de bits de información de una instrucción a la siguiente, y ser capaz de predecir (y aprovechar) el efecto de todo ese movimiento. Cuando se tenía toda esa información adherida a tu córtex cerebral, era casi como si tu propia mente se hubiera fusionado con el ambiente de computacional. A veces, se requerían horas para llegar al punto en que los pensamientos podían alcanzar esa amalgama total, y al llegar a tal punto, era una pena desperdiciarlo, así que intentaban sostenerlo, con ráfagas maratónicas, alternando entre trabajar en la computadora o estudiar el código escrito en una de las Flexowriters fuera de línea en la Sala Kluge. Mantenían dicha concentración "funcionando" hasta el día siguiente.

Inevitablemente, ese estado de fusión se extendía a los fragmentos aleatorios de existencia que los hackers tenían fuera de la informática. El contingente de espátulas y pinceles del TMRC no estaba nada contento con los infiltrados de la *Tixomanía* en el club: lo consideraban como una especie de caballo de Troya para alterar el foco del club, del ferrocarril a la informática. Y si uno asistía a una de las reuniones del club que se celebraban todos los martes a las cinco y cuarto, podía ver la conmoción: los hackers explotarían cada táctica posible del procedimiento parlamentario para crear una reunión tan enrevesada como los programas que estaban hackeando en el TX-0. Se hacían mociones para hacer mociones para hacer mociones, y las objeciones se descartaban como si fueran errores informáticos. Una nota en las actas de la reunión del 24 de noviembre de 1959 sugiere que "desaprobamos a ciertos miembros que harían mucho más bien al club si se dedicaran más a SyE y leyeran menos las Reglas de Orden de Robert". Samson era uno de los peores infractores, y en un momento dado, un miembro exasperado del TMRC hizo una moción "para comprar un corcho para la diarrea oral de Samson".

Hackear el procedimiento parlamentario era una cosa, pero el marco mental lógico necesario para la programación se extendía a actividades más comunes. Uno podía hacerle una pregunta a un hacker y sentir cómo su acumulador mental procesaba bits hasta que daba una respuesta precisa a la pregunta que uno le hacía. Marge Saunders iba todos los sábados por la mañana al Safeway en su Volkswagen y, al regresar, le preguntaba a su marido: "¿Te gustaría ayudarme a llevar las compras?". Bob Saunders respondía: "No". Atónita, Marge llevaba las compras ella misma. Después de que lo mismo sucediera varias veces, la esposa explotó, lanzándole maldiciones y exigiéndole que le dijera por qué había dicho que no a su pregunta.

"Es una pregunta estúpidamente realizada", dijo. "Por supuesto que *no me gustaría* ayudarte a llevar las compras. Si me preguntas *si te ayudaría* a llevarlas, eso es otro asunto".

Era como si Marge hubiera enviado un programa al TX-0 y este - como ocurre con los programas cuando la sintaxis es incorrecta - se hubiese bloqueado. No fue hasta que ella *depuró* su pregunta que Bob Saunders permitió que se ejecutara correctamente en su propia computadora mental.

Capítulo 2. La ética hacker

Algo nuevo se estaba formando alrededor del TX-0: una nueva forma de vida con una filosofía, una ética y un sueño.

No hubo un momento en el que los hackers del TX-0 comenzaran a darse cuenta de que al dedicar sus habilidades técnicas a la informática con una devoción que rara vez se ve fuera de los monasterios, eran la vanguardia de una audaz simbiosis entre el hombre y la máquina. Con un fervor como el de los jóvenes aficionados a los coches deportivos obsesionados con tunear motores, llegaron a dar por sentado su entorno casi único. Incluso cuando se estaban formando los elementos de una cultura, cuando comenzaron a acumularse leyendas, cuando su dominio de la programación comenzó a superar cualquier nivel de habilidad registrado anteriormente, la docena de hackers se mostraron reacios a reconocer que su pequeña sociedad, en íntimos términos con el TX-0, había estado juntando lenta e implícitamente un cuerpo de conceptos, creencias y costumbres.

Los preceptos de esta revolucionaria ética hacker no fueron tanto debatidos ni discutidos como acordados en silencio. No se emitieron manifiestos. Ningún misionero intentó reunir adeptos. La computadora fue la que convirtió, y quienes parecían seguir la ética hacker más fielmente eran personas como Samson, Saunders y Kotok, cuyas vidas antes del MIT parecían ser meros preludios de ese momento en que se realizaron detrás de la consola del TX-0. Más tarde vendrían hackers que se tomarían la ética implícita incluso más en serio que los hackers del TX-0, hackers como el legendario Greenblatt o Gosper, aunque pasarían algunos años antes de que los principios del hackerismo se delinearan explícitamente.

Aun así, incluso en los días del TX-0, los pilares de la plataforma estaban en su lugar. La ética hacker:

> El acceso a las computadoras -y a cualquier cosa que pueda enseñarte algo sobre cómo funciona el mundo- debe ser ilimitado y total. ¡Siempre cede ante el imperativo práctico!

Los hackers creen que pueden aprenderse lecciones esenciales sobre los sistemas (sobre el mundo) desmontando cosas, viendo cómo funcionan y usando este conocimiento para crear cosas nuevas e incluso más interesantes. Resienten a cualquier persona, barrera física o ley que intente impedirles hacerlo.

Esto es especialmente cierto cuando un hacker quiere arreglar algo que (desde su punto de vista) está roto o necesita mejorarse. Los sistemas imperfectos enfurecen a los hackers, cuyo instinto primario es depurarlos. Esta es una de las razones por las que los hackers generalmente odian conducir automóviles: el sistema de luces rojas programadas aleatoriamente y calles de un solo sentido con un diseño extraño causa demoras que son tan malditamente innecesarias que el impulso es reorganizar las señales, abrir las cajas de control de los semáforos... rediseñar todo el sistema.

En un mundo hacker perfecto, cualquiera que esté lo suficientemente enojado como para abrir una caja de control cerca de un semáforo y desmontarla para que funcione mejor debería ser perfectamente bienvenido a intentarlo. Las reglas que te impiden tomar ese tipo de asuntos en tus propias manos son demasiado ridículas para siquiera considerar cumplirlas. Esta actitud ayudó al Club del Ferrocarril Modelo a iniciar, de manera extremadamente informal, algo llamado el Comité de Requisición de Medianoche. Cuando el TMRC necesitaba un conjunto de diodos o algunos relés adicionales para incorporar alguna característica nueva al Sistema, algunas personas de SyE

esperaban hasta que oscureciera y encontraban el camino hacia los lugares donde se guardaban esas cosas. Ninguno de los hackers, que por lo general eran escrupulosamente honestos en otros asuntos, parecía equiparar esto con "robar". Una ceguera voluntaria.

Toda la información debería ser libre.

Si no tienes acceso a la información que necesitas para mejorar las cosas, ¿cómo puedes arreglarlas? Un libre intercambio de información, particularmente cuando la información estaba en forma de programa de computadora, permitía una mayor creatividad general. Cuando trabajabas en una máquina como la TX-0, que venía casi sin software, todos escribían frenéticamente programas de sistemas para facilitar la programación: Herramientas para hacer herramientas, guardadas en el cajón junto a la consola para que cualquiera que usara la máquina pudiera acceder a ellas fácilmente. Esto impidió el temido y derrochador ritual de reinventar la rueda: en lugar de que cada uno escribiera su propia versión del mismo programa, la mejor versión estaría disponible para todos, y todos serían libres de profundizar en el código y mejorarla. Un mundo plagado de programas repletos de funciones, modificados al mínimo, depurados a la perfección.

La creencia, a veces tomada incondicionalmente, de que la información debería ser libre era un tributo directo a la forma en que funciona una computadora o un programa de computadora espléndidos: los bits binarios se mueven en el camino más directo y lógico necesario para hacer su complejo trabajo. ¿Qué era una computadora sino algo que se beneficiaba de un flujo libre de información? Si, por ejemplo, el acumulador se encontraba incapaz de obtener información de los dispositivos de entrada/salida (E/S) como el lector de cintas o los conmutadores, todo el sistema colapsaría. Desde el punto de vista del hacker, cualquier sistema podría beneficiarse de ese flujo fácil de información.

Desconfíe de la autoridad: promueva la descentralización.

La mejor manera de promover este intercambio libre de la información es contar con un sistema abierto, algo que no presente límites entre un hacker y una pieza de información o un elemento de equipo que necesita en su búsqueda de conocimiento, mejora y tiempo en línea. Lo último que se necesita es una burocracia. Las burocracias, ya sean corporativas, gubernamentales o universitarias, son sistemas defectuosos, peligrosos porque no pueden dar cabida al impulso exploratorio de los verdaderos hackers. Los burócratas se esconden detrás de reglas arbitrarias (en contraposición a los algoritmos lógicos por los que funcionan las máquinas y los programas informáticos): invocan esas reglas para consolidar el poder y perciben el impulso constructivo de los hackers como una amenaza.

El epítome del mundo burocrático se encontraba en una empresa muy grande llamada International Business Machines, IBM. La razón por la que sus computadoras eran gigantes enormes procesados por lotes se debía sólo en parte a la tecnología de tubos de vacío. La verdadera razón era que IBM era una empresa torpe y descomunal que no entendía el impulso del hacker. Si IBM se saliera con la suya (eso pensaban los hackers del TMRC), el mundo se procesaría por lotes, se distribuiría en esas molestas tarjetas perforadas, y sólo los sacerdotes más privilegiados tendrían permitido interactuar con la computadora.

Todo lo que había que hacer era mirar a alguien en el mundo de IBM y notar la camisa blanca abotonada, la corbata negra prolijamente sujeta con alfileres, el cabello cuidadosamente recogido y la bandeja de tarjetas perforadas en la mano. Se podía pasear por el Centro de Cálculo, donde se

almacenaban el 704, el 709 y más tarde el 7090 (lo mejor que IBM tenía para ofrecer) y ver el orden agobiante, hasta las áreas acordonadas más allá de las cuales las personas no autorizadas no podían aventurarse. Y se podía comparar eso con la atmósfera extremadamente informal alrededor del TX-0, donde la ropa sucia era la norma y casi cualquiera podía entrar. Ahora bien, IBM había hecho y seguiría haciendo muchas cosas para hacer avanzar la informática. Por su gran tamaño y poderosa influencia, había convertido a las computadoras en una parte permanente de la vida en Estados Unidos. Para mucha gente, las palabras “IBM” y “ordenador” eran prácticamente sinónimos. Las máquinas de IBM eran caballos de batalla fiables, dignos de la confianza que los empresarios y los científicos depositaban en ellas. Esto se debía en parte al enfoque conservador de IBM: no fabricaría las máquinas tecnológicamente más avanzadas, sino que se basaría en conceptos probados y en una comercialización cuidadosa y agresiva. A medida que IBM se consolidaba en el campo de los ordenadores, la empresa se convirtió en un imperio en sí misma, reservada y presumida.

Lo que realmente volvía locos a los hackers era la actitud de los sumo sacerdotes y sacerdotes de IBM, que parecían pensar que IBM tenía los únicos ordenadores “reales” y que el resto eran basura. No se podía hablar con esa gente; eran más que convincentes. Eran gente que procesaba todo por lotes, y eso se notaba no sólo en su preferencia por las máquinas, sino en sus ideas sobre cómo debería gestionarse un centro de computación y un mundo. Esas personas nunca pudieron entender la obvia superioridad de un sistema descentralizado, sin nadie que diera órdenes, un sistema en el que la gente pudiera seguir sus intereses y, si en el camino descubriera un fallo en el sistema, pudiera embarcarse en una ambiciosa cirugía. No había necesidad de conseguir un formulario de solicitud. Sólo la necesidad de hacer algo.

Esta inclinación antiburocrática coincidía perfectamente con las personalidades de muchos de los hackers, que desde la infancia se habían acostumbrado a desarrollar proyectos científicos mientras el resto de sus compañeros de clase se golpeaban las cabezas y aprendían habilidades sociales en el campo de juego. Estos jóvenes adultos que alguna vez fueron parias encontraron en la computadora un fantástico ecualizador, experimentando una sensación, según Peter Samson, "como si abrieras la puerta y caminaras por este gran universo nuevo..." Una vez que pasaban por esa puerta y se sentaban detrás de la consola de una computadora de un millón de dólares, los hackers tenían poder. Por lo tanto, era natural desconfiar de cualquier fuerza que pudiera intentar limitar el alcance de ese poder.

> Los hackers deberían ser juzgados por sus habilidades, no por criterios falsos como títulos, edad, raza o posición.

La pronta aceptación de Peter Deutsch, de doce años, en la comunidad TX-0 (aunque no por parte de estudiantes de posgrado que no fueran hackers) fue un buen ejemplo. De la misma manera, a las personas que llegaban con credenciales aparentemente impresionantes no se las tomaba en serio hasta que demostraban su valía frente a la consola de una computadora. Este rasgo meritocrático no tenía necesariamente su raíz en la bondad inherente de los corazones de los hackers; se debía principalmente a que a los hackers les importaban menos las características superficiales de alguien que su potencial para hacer avanzar el estado general del hacking, para crear nuevos programas para admirar, para hablar sobre esa nueva característica del sistema.

> Se puede crear arte y belleza en una computadora.

El programa de música de Samson fue un ejemplo. Pero para los hackers, el arte del programa no residía en los sonidos agradables que emanaban del altavoz en línea. El código del programa tenía una belleza propia. (Samson, sin embargo, fue particularmente oscuro al negarse a agregar comentarios a su código fuente explicando lo que estaba haciendo en un momento dado. Un programa bien distribuido que escribió Samson se extendía por cientos de instrucciones en lenguaje ensamblador, con sólo un comentario al lado de una instrucción que contenía el número 1750. El comentario era RIPJSB, y la gente se devanó los sesos sobre su significado hasta que alguien descubrió que 1750 era el año en que murió Bach, y que Samson había escrito una abreviatura de Descanse en paz Johann Sebastian Bach.)

Había surgido una cierta estética del estilo de programación. Debido al espacio de memoria limitado del TX-0 (una desventaja que se extendió a todos los ordenadores de esa época), los hackers llegaron a apreciar profundamente las técnicas innovadoras que permitían a los programas realizar tareas complicadas con muy pocas instrucciones. Cuanto más corto era un programa, más espacio quedaba para otros programas y más rápido se ejecutaba. A veces, cuando no necesitabas mucha velocidad o espacio, y no estabas pensando en el arte y la belleza, armabas un programa feo, atacando el problema con métodos de "fuerza bruta". “Bien, podemos hacer esto sumando veinte números”, podría decirse Samson, “y es más rápido escribir instrucciones para hacer eso que pensar en un bucle al principio y al final para hacer el mismo trabajo en siete u ocho instrucciones”. Pero este último programa podría ser admirado por otros hackers, y algunos programas fueron modificados hasta el mínimo de líneas con tanta habilidad que los colegas del autor lo mirarían y casi se derretirían de asombro.

A veces, la modificación de programas se convertía en una competencia, una competencia de machos para demostrar que uno dominaba tanto el sistema que podía reconocer atajos elegantes para quitar una instrucción o dos, o, mejor aún, replantear todo el problema y diseñar un nuevo algoritmo que ahorrara un bloque entero de instrucciones. (Un algoritmo es un procedimiento específico que se puede aplicar para resolver un problema informático complejo; es una especie de llave maestra matemática). Esto se podía hacer con mayor énfasis abordando el problema desde un ángulo poco convencional en el que nadie había pensado antes, pero que en retrospectiva tenía todo el sentido. Definitivamente, había un impulso artístico en quienes podían utilizar esta técnica de genio de Marte: una cualidad visionaria de magia negra que les permitía descartar la perspectiva rancia de las mejores mentes de la Tierra y crear un algoritmo totalmente inesperado.

Esto sucedió con el programa de rutina de impresión decimal. Se trataba de una subrutina (un programa dentro de un programa que a veces se podía integrar en muchos programas diferentes) para traducir los números binarios que el ordenador te daba a números decimales regulares. En palabras de Saunders, este problema se convirtió en el «culo del peón de la programación: si podías escribir una rutina de impresión decimal que funcionara, sabías lo suficiente sobre el ordenador para llamarte una especie de programador». Y si escribías una gran rutina de impresión decimal, tal vez pudieras llamarte un hacker. Más que una competición, la máxima decepción de la rutina de impresión decimal se convirtió en una especie de Santo Grial de los hackers.

Hacía algunos meses que existían varias versiones de rutinas de impresión decimal. Si uno se comportaba deliberadamente como un estúpido, o si era un auténtico imbécil (un «perdedor» de pura cepa), podría necesitar cien instrucciones para conseguir que el ordenador convirtiera el lenguaje de máquina a decimal. Pero cualquier hacker que se precie podía hacerlo en menos tiempo

y, finalmente, tomando los mejores programas y copiando una instrucción aquí y allá, la rutina se redujo a unas cincuenta instrucciones.

Después de eso, las cosas se pusieron serias. La gente trabajaba durante horas, buscando una forma de hacer lo mismo en menos líneas de código. Se convirtió en algo más que una competición; era una búsqueda. A pesar de todo el esfuerzo invertido, nadie parecía ser capaz de superar la barrera de las cincuenta líneas. Surgió la pregunta de si era posible hacerlo en menos tiempo. ¿Había un punto más allá del cual no se podía copiar un programa? Entre las personas que se debatían con este dilema se encontraba un tipo llamado Jensen, un hacker alto y silencioso de Maine que se sentaba tranquilamente en la Sala Kluge y garabateaba en las impresiones con la actitud tranquila de un hombre de los bosques que talla una piedra. Jensen siempre estaba buscando formas de comprimir sus programas en el tiempo y el espacio: su código era una secuencia completamente extraña de funciones booleanas y aritméticas entremezcladas, que a menudo hacía que se produjeran varios cálculos diferentes en diferentes secciones de la misma "palabra" de dieciocho bits. Cosas asombrosas, trucos mágicos.

Antes de Jensen, había un acuerdo general de que el único algoritmo lógico para una rutina de impresión decimal haría que la máquina restara repetidamente, utilizando una tabla de potencias de diez para mantener los números en las columnas digitales adecuadas. De alguna manera, Jensen se dio cuenta de que una tabla de potencias de diez no era necesaria; se le ocurrió un algoritmo que era capaz de convertir los dígitos en orden inverso, pero, mediante algún truco de magia digital, imprimirlos en el orden adecuado. Había una justificación matemática compleja que quedó clara para los otros hackers solo cuando vieron el programa de Jensen publicado en un tablón de anuncios, su forma de dicirles que había llevado la rutina de impresión decimal hasta el límite. Cuarenta y seis instrucciones. La gente miraba el código y se quedaba boquiabierta. Marge Saunders recuerda que los hackers estuvieron inusualmente callados durante días después.

“Sabíamos que era el final”, dijo Bob Saunders más tarde. “Eso fue el Nirvana”.

> Las computadoras pueden cambiar tu vida para mejor.

Esta creencia se manifestaba sutilmente. Rara vez un hacker intentaría imponer una visión de las innumerables ventajas del modo de conocimiento informático a un extraño. Sin embargo, esta premisa dominaba el comportamiento cotidiano de los hackers TX-0, así como de las generaciones de hackers que vinieron después de ellos.

Seguramente la computadora había cambiado sus vidas, las había enriquecido, les había dado un enfoque, las había vuelto aventureras. Los había convertido en dueños de una cierta porción del destino. Peter Samson dijo más tarde: “Lo hicimos en un veinticinco o treinta por ciento por el mero hecho de hacerlo, porque era algo que podíamos hacer y hacer bien, y en un sesenta por ciento por el hecho de tener algo que, en cierto modo metafórico, estuviera vivo, nuestra descendencia, que haría las cosas por sí sola cuando termináramos. Eso es lo bueno de la programación, el atractivo mágico que tiene... Una vez que se soluciona un problema de comportamiento [de una computadora o programa], se soluciona para siempre, y es exactamente una imagen de lo que uno quería decir”.

Al igual que la lámpara de Aladino, uno podía hacer que hiciera lo que uno quería.

Seguramente todos podrían beneficiarse de experimentar este poder. Seguramente todos podrían beneficiarse de un mundo basado en la Ética Hacker. Esta era la creencia implícita de los hackers, y

los hackers extendieron irreverentemente el punto de vista convencional de lo que las computadoras podían y debían hacer, llevando al mundo a una nueva forma de ver e interactuar con las computadoras.

Esto no era fácil de hacer. Incluso en una institución tan avanzada como el MIT, algunos profesores consideraban que una afinidad maníaca por los ordenadores era frívola, incluso demente. El hacker del TMRC Bob Wagner tuvo que explicar una vez a un profesor de ingeniería qué era un ordenador. Wagner experimentó este choque entre ordenador y antiordenador de forma aún más vívida cuando tomó una clase de Análisis Numérico en la que el profesor exigía a cada alumno que hiciera los deberes utilizando calculadoras electromecánicas ruidosas y torpes. Kotok estaba en la misma clase, y a ambos les horrorizaba la perspectiva de trabajar con esas máquinas de baja tecnología. “¿Por qué deberíamos hacerlo”, preguntaban, “si tenemos este ordenador?”.

Así que Wagner empezó a trabajar en un programa informático que emulase el comportamiento de una calculadora. La idea era escandalosa. Para algunos, era una apropiación indebida del valioso tiempo de la máquina. Según el pensamiento estándar sobre los ordenadores, su tiempo era tan precioso que uno sólo debía intentar cosas que sacaran el máximo partido del ordenador, cosas que de otro modo llevarían días enteros de cálculos sin sentido a salas llenas de matemáticos. Los hackers pensaban de otra manera: todo lo que pareciera interesante o divertido era material para la computación, y usando computadoras interactivas, sin nadie que te mirara por encima del hombro y te exigiera permiso para tu proyecto específico, podías actuar de acuerdo con esa creencia. Después de dos o tres meses de enredarse con las complejidades de la aritmética de punto flotante (necesaria para que el programa sepa dónde colocar el punto decimal) en una máquina que no tenía un método simple para realizar multiplicaciones elementales, Wagner había escrito tres mil líneas de código que hacían el trabajo. Había logrado que una computadora ridículamente cara realizara la función de una calculadora que costaba una milésima parte del precio. Para honrar esta ironía, llamó al programa "Calculadora de escritorio costosa" y orgullosamente hizo la tarea para su clase en él.

Su calificación: cero. "¡Usaste una computadora!", le dijo el profesor. "Esto no puede ser correcto".

Wagner ni siquiera se molestó en explicarlo. ¿Cómo podía transmitirle a su profesor que el ordenador estaba haciendo realidades a partir de posibilidades que antes eran increíbles? ¿O que otro hacker había escrito incluso un programa llamado “Máquina de escribir cara” que convertía el TX-0 en algo en lo que se podía escribir texto, podía procesar tu escritura en cadenas de caracteres e imprimirla en la Flexowriter? ¿Se imaginaba a un profesor aceptando un informe de clase escrito por el ordenador? ¿Cómo podía ese profesor (cómo podía, de hecho, cualquiera que no hubiera estado inmerso en ese universo inexplorado del hombre y la máquina) entender que Wagner y sus compañeros hackers utilizaban rutinariamente el ordenador para simular, según Wagner, “situaciones extrañas que uno difícilmente podría imaginar de otra manera”? El profesor aprendería con el tiempo, como todos, que el mundo abierto por el ordenador era ilimitado.

Si alguien necesitaba más pruebas, se podía citar el proyecto en el que trabajaba Kotok en el Centro de Cálculo, el programa de ajedrez que llevaba el profesor de IA “Tío” John McCarthy, como lo estaban conociendo sus estudiantes hackers, había comenzado en el IBM 704. A pesar de que Kotok y los otros hackers que lo ayudaban en el programa solo sentían desprecio por la mentalidad de procesamiento por lotes de IBM que impregnaba la máquina y la gente que lo rodeaba había logrado encontrar algo de tiempo a altas horas de la noche para usarlo de manera interactiva y había estado enfrascado en una batalla informal con los programadores de sistemas del 704 para ver qué

grupo sería conocido como el mayor consumidor de tiempo de computadora. El liderazgo iba y venía, y la gente de camisa blanca y corbata negra del 704 quedó lo suficientemente impresionada como para dejar que Kotok y su grupo tocaran los botones e interruptores del 704: un contacto sensual poco común con una bestia de IBM alabada.

El papel de Kotok en dar vida al programa de ajedrez fue indicativo de lo que se convertiría en el papel del hacker en la Inteligencia Artificial: un Cabeza Dura como McCarthy o su colega Marvin Minsky comenzaría un proyecto o se preguntaría en voz alta si algo podría ser posible, y los hackers, si les interesaba, se pondrían a hacerlo.

El programa de ajedrez se había iniciado utilizando FORTRAN, uno de los primeros lenguajes informáticos. Los lenguajes informáticos se parecen más al inglés que al lenguaje ensamblador, son más fáciles de escribir y hacen más cosas con menos instrucciones; Sin embargo, cada vez que se da una instrucción en un lenguaje de programación como FORTRAN, la computadora debe traducir primero esa orden a su propio lenguaje binario. Un programa llamado “compilador” se encarga de esto, y el compilador requiere tiempo para hacer su trabajo, además de ocupar un espacio valioso dentro de la computadora. En efecto, usar un lenguaje de programación te aleja un paso más del contacto directo con la computadora, y los hackers generalmente preferían el lenguaje ensamblador o, como lo llamaban, el lenguaje “máquina” a lenguajes menos elegantes, de “alto nivel”, como FORTRAN.

Kotok, sin embargo, reconoció que debido a las enormes cantidades de números que tendrían que procesarse en un programa de ajedrez, parte del programa tendría que hacerse en FORTRAN y parte en ensamblador. Lo hackearon parte por parte, con “generadores de movimientos”, estructuras de datos básicas y todo tipo de algoritmos innovadores para la estrategia. Después de alimentar a la máquina con las reglas para mover cada pieza, le dieron algunos parámetros con los que evaluar su posición, considerar varios movimientos y hacer el movimiento que la llevaría a la situación más ventajosa. Kotok siguió trabajando en ello durante años, y el programa fue creciendo a medida que el MIT iba actualizando sus ordenadores IBM. Una noche memorable, unos cuantos hackers se reunieron para ver cómo el programa hacía algunos de sus primeros movimientos en una partida real. Su primer movimiento fue bastante respetable, pero después de unos ocho intercambios hubo un verdadero problema, ya que el ordenador estaba a punto de recibir jaque mate. Todo el mundo se preguntaba cómo reaccionaría el ordenador. Tardó un rato (todo el mundo sabía que durante esas pausas el ordenador estaba realmente "pensando", si la idea de pensar incluía considerar mecánicamente varios movimientos, evaluarlos, rechazar la mayoría y utilizar un conjunto predefinido de parámetros para finalmente tomar una decisión). Finalmente, el ordenador avanzó un peón dos casillas, saltando ilegalmente sobre otra pieza. ¡Un error! Pero uno inteligente: sacó al ordenador del jaque. Tal vez el programa estaba ideando un nuevo algoritmo con el que conquistar el ajedrez.

En otras universidades, los profesores hacían proclamaciones públicas de que los ordenadores nunca serían capaces de vencer a un ser humano en ajedrez. Los hackers lo sabían mejor. Serían ellos los que llevarían a los ordenadores a mayores alturas de las que nadie esperaba. Y los hackers, por su fructífera y significativa asociación con la computadora, serían los principales beneficiarios. Pero no serían los únicos beneficiarios. Todo el mundo podría ganar algo con el uso de computadoras pensantes en un mundo intelectualmente automatizado. ¿Y no se beneficiarían todos aún más si abordaran el mundo con la misma intensidad inquisitiva, escepticismo hacia la burocracia, apertura a la creatividad, altruismo al compartir logros, impulso a realizar mejoras y

deseo de construir que aquellos que seguían la Ética Hacker? ¿Aceptando a los demás con la misma base imparcial con la que las computadoras aceptaban a cualquiera que ingresara un código en una Flexowriter? ¿No nos beneficiaríamos si aprendiéramos de las computadoras los medios para crear un sistema perfecto y nos propusiéramos emular esa perfección en un sistema humano? Si todos pudiéramos interactuar con las computadoras con el mismo impulso inocente, productivo y creativo que los hackers, la Ética Hacker podría extenderse por la sociedad como una onda benévola, y las computadoras realmente cambiarían el mundo para mejor.

En los confines monásticos del Instituto Tecnológico de Massachusetts, la gente tenía la libertad de vivir este sueño: el sueño hacker. Nadie se atrevió a sugerir que el sueño pudiera propagarse. En cambio, la gente se dedicó a construir, allí mismo en el MIT, un Xanadu hacker, algo que tal vez nunca se pueda duplicar.

Capítulo 3. Spacewar

En el verano de 1961, Alan Kotok y los demás hackers del TMRC se enteraron de que una nueva empresa iba a entregar pronto al MIT, absolutamente gratis, el siguiente paso en informática, una máquina que llevaría los principios interactivos del TX-0 varios pasos más allá. Una máquina que podría ser incluso mejor para los hackers que el TX-0.

El PDP-1. Cambiaría la informática para siempre. Haría que el todavía confuso sueño hacker se volviera un poco más cercano a la realidad.

Alan Kotok se había distinguido como un verdadero mago del TX-0, tanto que él, junto con Saunders, Samson, Wagner y algunos otros, habían sido contratados por Jack Dennis para ser el Grupo de Programación de Sistemas del TX-0. El salario sería de unos generosos 1,60 dólares la hora. Para algunos de los hackers, el trabajo era una excusa más para no ir a clases: algunos hackers, como Samson, nunca se graduarían y estarían demasiado ocupados hackeando como para lamentar realmente la pérdida. Kotok, sin embargo, no sólo era capaz de gestionar sus clases, sino de establecerse como un hacker "canónico". En el TX-0 y el TMRC, estaba adquiriendo un estatus legendario. Un hacker que acababa de llegar al MIT ese año recuerda a Kotok dando a los recién llegados una demostración de cómo funcionaba el TX-0: "Me dio la impresión de que era hipertiroideo o algo así", recordaba Bill Gosper, que se convertiría en un hacker canónico, "porque hablaba muy despacio y era rechoncho y tenía los ojos entrecerrados. Esa era una impresión completamente errónea. [En el TX-0] Kotok tenía una autoridad moral infinita. Había escrito el programa de ajedrez. Entendía el hardware". (Esto último no era un cumplido desdeñable: "entender el hardware" era similar a comprender el Tao de la naturaleza física).

El verano en que se supo del PDP-1, Kotok estaba trabajando para Western Electric, una especie de trabajo de ensueño, ya que de todos los sistemas posibles, el sistema telefónico era el que más admiraba. El Club de Ferrocarriles Modelo solía hacer visitas guiadas a las centrales telefónicas de las compañías, de forma muy similar a como lo harían las personas interesadas en la pintura cuando visitan un museo. A Kotok le pareció interesante que en la compañía telefónica, que había crecido tanto en sus décadas de desarrollo, sólo unos pocos ingenieros tuvieran un amplio conocimiento de las interrelaciones dentro de ese sistema. Sin embargo, los ingenieros podían proporcionar fácilmente detalles sobre funciones específicas del sistema, como la conmutación de barras cruzadas y los relés de paso; Kotok y los demás acosaban a estos expertos para que les dieran información, y los ingenieros halagados, probablemente sin tener idea de que esos universitarios ultraeducados realmente usarían la información, accedían de inmediato.

Kotok se propuso asistir a esas visitas, leer todo el material técnico que pudiera conseguir y ver qué podía obtener marcando diferentes números en el complejo y poco comprendido sistema telefónico del MIT. Era una exploración básica, como explorar los callejones digitales del TX-0. Durante el invierno anterior de 1960-61, los hackers del TMRC habían llevado a cabo una elaborada «huella digital de la red telefónica», trazando un mapa de todos los lugares a los que se podía acceder mediante el sistema de líneas de enlace del MIT. Aunque no estuviera conectado a las líneas telefónicas generales, el sistema podía llevarte al Laboratorio Lincoln y, desde allí, a los contratistas de defensa de todo el país. Era cuestión de mapear y probar. Empezabas con un código de acceso, le añadías distintos dígitos, veías quién podía responder, preguntabas a quien respondiera dónde estaba y luego añadías dígitos a ese número para ir directamente al siguiente lugar. A veces incluso podías

llegar a líneas externas en los suburbios, cortesía de la desprevenida compañía telefónica. Y, como Kotok admitiría más tarde, «si había algún fallo de diseño en el sistema telefónico que permitiera recibir llamadas que no estaban destinadas a pasar, yo no tenía reparos en hacerlo, pero ese era su problema, no el mío».

Aun así, el motivo era la exploración, no el fraude, y se consideraba de mala educación sacar provecho ilegal de estas extrañas conexiones. A veces, los extraños no podían comprenderlo. Los compañeros de habitación de Samson en la residencia de estudiantes de Burton Hall, por ejemplo, no eran hackers y pensaban que estaba bien explotar los errores del sistema sin la justificación sagrada de la exploración del sistema. Después de presionar a Samson durante días, finalmente cedió y les dio un número de 20 dígitos que, según dijo, les permitiría acceder a una ubicación exótica. “Pueden marcar esto desde el teléfono del pasillo”, les dijo, “pero no quiero estar cerca”. Mientras empezaban a marcar ansiosamente, Samson se dirigió a un teléfono de la planta baja, que sonó justo cuando él lo alcanzaba. “Este es el Pentágono”, gritó con su voz más oficial. “¿Cuál es su autorización de seguridad, por favor?” Desde el teléfono de arriba, Samson escuchó jadeos aterrorizados y el clic de un teléfono que colgaba.

La toma de huellas digitales de la red era obviamente una actividad limitada a los hackers, cuyo deseo de conocer el sistema superaba cualquier miedo a ser atrapados.

Pero por mucho que el esoterismo de las compañías telefónicas fascinara a Kotok, la perspectiva del PDP-1 tenía prioridad. Tal vez intuyó que nada, ni siquiera el espionaje telefónico, volvería a ser lo mismo después. Las personas que diseñaron y comercializaron estas nuevas máquinas no eran las típicas computadoras de empresa. La empresa era una firma nueva llamada Digital Equipment Corporation (DEC), y algunos de los usuarios de TX-0 sabían que los primeros productos de DEC eran interfaces especiales hechas específicamente para esa TX-0. Era bastante emocionante que algunos de los fundadores de DEC tuvieran una visión de la informática que difería de la mentalidad de IBM, de franela gris y procesamiento por lotes; era realmente asombroso que la gente de DEC pareciera haber observado el estilo libre, interactivo, improvisado y práctico de la comunidad TX-0, y diseñado una computadora que reforzara ese tipo de comportamiento. La PDP-1 (las iniciales eran la abreviatura de Programmed Data Processor, un término considerado menos amenazador que “computadora”, que tenía todo tipo de connotaciones de gigante descomunal) llegaría a ser conocida como la primera minicomputadora, diseñada no para tareas de procesamiento de números enormes, sino para la investigación científica, la formulación matemática... y el hackeo. Sería tan compacto que el conjunto no ocuparía más que tres frigoríficos, no necesitaría tanto aire acondicionado y hasta se podría poner en marcha sin que fuera necesario un equipo entero de sub-sacerdotes para secuenciar varias fuentes de alimentación en el orden correcto o poner en marcha el generador de base de tiempo, entre otras tareas exigentes. El precio de venta al público del ordenador era de unos sorprendentemente bajos 120.000 dólares, lo bastante barato como para que la gente dejara de quejarse de lo precioso que era cada segundo de tiempo de ordenador. Pero la máquina, que era la segunda PDP-1 fabricada (la primera se vendió a la cercana empresa científica Bolt Beranek and Newman, o BBN), no le costó nada al MIT: fue donada por el DEC al laboratorio RLE.

Así que estaba claro que los hackers tendrían incluso más tiempo en ella que en la TX-0.

La PDP-1 se entregaría con una sencilla colección de software de sistemas, que los hackers consideraban completamente inadecuado. Los hackers del TX-0 se habían acostumbrado al software

interactivo más avanzado del mundo, un conjunto deslumbrante de programas de sistemas escritos por los propios hackers y adaptados implícitamente a sus incesantes demandas de control de la máquina. El joven Peter Deutsch, el niño de doce años que había descubierto el TX-0, había cumplido su promesa de escribir un ensamblador más elegante, y Bob Saunders había desarrollado una versión más pequeña y rápida del depurador FLIT llamado Micro-FLIT. Estos programas se habían beneficiado de un conjunto de instrucciones ampliado. Un día, después de una considerable planificación y diseño por parte de Saunders y Jack Dennis, el TX-0 se había apagado y un grupo de ingenieros expuso sus entrañas y comenzó a cablear nuevas instrucciones en la máquina. Esta formidable tarea amplió el lenguaje ensamblador en varias instrucciones. Cuando se guardaron los alicates y los destornilladores y se encendió cuidadosamente el ordenador, todos se pusieron como locos a renovar programas y a deshacer programas antiguos utilizando las nuevas instrucciones. Kotok se enteró de que el conjunto de instrucciones del PDP-1 no era muy diferente del del TX-0 ampliado, por lo que Kotok comenzó naturalmente a escribir software de sistemas para el PDP-1 ese mismo verano, utilizando todo el tiempo libre que pudo conseguir. Suponiendo que todo el mundo se pondría manos a la obra y empezaría a escribir en cuanto llegase la máquina, trabajó en una traducción del depurador Micro-FLIT para que escribir el software para el “One” fuese más fácil. Samson no tardó en bautizar el depurador de Kotok como “DDT”, y el nombre se mantuvo, aunque el programa en sí mismo sería modificado incontables veces por hackers que querían añadirle funciones o extraerle instrucciones.

Kotok no era el único que se preparaba para la llegada del PDP-1. Como una variopinta colección de futuros padres, otros hackers estaban ocupados tejiendo patucos y mantas de software para el nuevo bebé que llegaba a la familia, por lo que este anunciado heredero al trono de la informática sería bienvenido tan pronto como llegase a finales de septiembre.

Los hackers ayudaron a llevar el PDP-1 a su nuevo hogar, la Sala Kluge, al lado del TX-0. Era una belleza: sentado detrás de una consola de la mitad de longitud que la del Tixo, veías un panel compacto de interruptores y luces; al lado estaba la pantalla, encerrada en una carcasa azul brillante, de seis lados, casi de tipo sideco; detrás estaban los armarios altos, del tamaño de un frigorífico y tres veces más profundos, con los cables, las placas, los interruptores y los transistores (la entrada a ellos, por supuesto, estaba prohibida). Había una Flexowriter conectada para la entrada en línea (la gente se quejó tanto del ruido que la Flexowriter acabó siendo sustituida por una máquina de escribir IBM modificada, que no funcionaba ni de lejos tan bien) y un lector de cinta de papel de alta velocidad, también para la entrada. En conjunto, un juguete absolutamente celestial.

A Jack Dennis le gustó parte del software escrito por BBN para el prototipo PDP-1, en particular el ensamblador. Sin embargo, Kotok sintió náuseas cuando vio que se ejecutaba el ensamblador (el modo de operación no parecía adaptarse al estilo sobre la marcha que le gustaba), por lo que él y algunos otros le dijeron a Dennis que querían escribir el suyo propio. "Esa es una mala idea", dijo Dennis, que quería un ensamblador en marcha.

El programa se puso en marcha de inmediato y se calculó que los hackers tardarían semanas en hacerlo.

Kotok y los demás se mostraron inflexibles. Era un programa con el que tendrían que vivir. Tenía que ser perfecto. (Por supuesto, ningún programa lo es, pero eso nunca detuvo a un hacker).

“Te diré una cosa”, le dijo Kotok, este mago con forma de Buda de veinte años, al escéptico pero comprensivo Jack Dennis, “si escribimos este programa durante el fin de semana y lo hacemos funcionar, ¿nos pagarías por el tiempo?”.

La escala salarial en ese momento era tal que el total sería algo menos de quinientos dólares. “Eso parece un trato justo”, dijo Dennis.

Kotok, Samson, Saunders, Wagner y un par más comenzaron un viernes por la noche a fines de septiembre. Calcularon que trabajarían con el ensamblador TX-0 cuyo original Dennis había escrito y que Peter Deutsch, de doce años, entre otros, había renovado. No cambiarían las entradas ni las salidas, y no rediseñarían los algoritmos; cada hacker tomaría una sección del programa TX-0 y la convertiría en código PDP-1. Y no dormirían. Seis hackers trabajaron alrededor de doscientas cincuenta horas-hombre ese fin de semana, escribiendo código, depurando y comiendo comida china para llevar con cantidades masivas de Coca-Cola enviadas desde la sala del club TMRC. Fue una orgía de programación, y cuando Jack Dennis llegó ese lunes, se sorprendió al encontrar un ensamblador cargado en el PDP-1, que, como demostración, estaba ensamblando su propio código en binario.

A fuerza de hackeo, los hackers TX-0 —no, PDP-1— habían desarrollado un programa en un fin de semana que a la industria informática le habría llevado semanas, tal vez incluso meses, llevarlo a cabo. Era un proyecto que probablemente la industria informática no emprendería sin un largo y tedioso proceso de solicitudes, estudios, reuniones y vacilaciones ejecutivas, muy probablemente con considerables concesiones en el camino. Tal vez nunca se hubiera llevado a cabo. El proyecto fue un triunfo para la ética hacker.

Los hackers obtuvieron incluso más acceso a esta nueva máquina que el que habían logrado obtener en la TX-0, y casi todos trasladaron sus operaciones a la sala Kluge. Unos pocos se mantuvieron tercamente en la Tixo, y para los hackers de la PDP-1, esto fue motivo de burla. Para restregarles la cabeza, los hackers de la PDP-1 desarrollaron una pequeña demostración basada en la mnemotecnia del conjunto de instrucciones de esta nueva y audaz máquina, que incluía instrucciones tan exóticas como DAC (Deposit Accumulator), LIO (Load Input-Output), DPY (Deplay) y JMP. El grupo de la PDP-1 se ponía en fila y gritaba al unísono:

LAC,

DAC,

DIPPY DAP,

LIO,

DIO

JUMP!

Cuando coreaban esa última palabra —“¡Salten!”— todos saltaban hacia la derecha. Lo que faltaba en la coreografía se compensaba con creces con el entusiasmo: estaban sobrealimentados por la belleza de la máquina, por la belleza de las computadoras.

El mismo tipo de entusiasmo era evidente en la programación aún más espontánea que se producía en el PDP-1, que abarcaba desde programas de sistemas serios hasta programas para controlar un brazo robótico primitivo y trucos extravagantes. Uno de estos últimos se aprovechó de una conexión hackeada entre el PDP-1 y el TX-0, un cable a través del cual podía pasar información, bit a bit, entre las dos máquinas. Según Samson, los hackers llamaron al venerable pionero de la IA John McCarthy para que se sentara junto al PDP-1. «Profesor McCarthy, ¡mire nuestro nuevo programa de ajedrez!». Y luego llamaron a otro profesor para que se sentara junto al TX-0. «¡Aquí está el programa de ajedrez! ¡Escriba su movimiento!». Después de que McCarthy escribiera su primer movimiento y este apareciera en la Flexowriter del TX-0, los hackers le dijeron al otro profesor que acababa de presenciar el movimiento inicial del TX-0. "¡Ahora haz el tuyo!" Después de unos cuantos movimientos, McCarthy se dio cuenta de que el ordenador estaba dando salida a los movimientos letra por letra, a veces con una pausa sospechosa entre ellos. Así que McCarthy siguió el hilo hasta su oponente de carne y hueso. Los hackers se reían de alegría. Pero no pasaría mucho tiempo antes de que se les ocurrieran programas para ordenadores (no es broma) para jugar realmente al ajedrez de torneo.

El PDP-1 atraía a los hackers a programar sin límites. Samson estaba hackeando cosas como el calendario maya (que funcionaba con un sistema numérico de base 20) y trabajando horas extra en una versión de su programa musical TX-0 que aprovechaba las capacidades de audio ampliadas del PDP-1 para crear música a tres voces: fugas de Bach a tres voces, melodías que interactuaban... ¡música de ordenador que brotaba de la antigua sala Kluge! La gente de DEC había oído hablar del programa de Samson y le pidió que lo completara en el PDP-1, así que Samson finalmente lo hizo funcionar de manera que alguien pudiera escribir una partitura musical en la máquina mediante una simple traducción de notas a letras y dígitos, y la computadora respondería con una sonata de órgano de tres voces. Otro grupo codificó operetas de Gilbert y Sullivan.

Samson presentó con orgullo el compilador de música a DEC para que lo distribuyera a cualquiera que lo quisiera. Estaba orgulloso de que otras personas usaran su programa. El equipo que trabajó en el nuevo ensamblador se sentía igual. Por ejemplo, les alegraba tener una cinta de papel con el programa en el cajón para que cualquiera que usara la máquina pudiera acceder a ella, intentar mejorarla, copiarle algunas instrucciones o añadirle alguna característica. Se sentían honrados cuando DEC les pidió el programa para poder ofrecérselo a otros propietarios de PDP-1. La cuestión de las regalías nunca surgió. Para Samson y los demás, utilizar el ordenador era una alegría tal que habrían pagado por hacerlo. El hecho de que les pagaran la principesca suma de 1,60 dólares la hora por trabajar en el ordenador era un plus. En cuanto a las regalías, ¿no era el software más bien un regalo al mundo, algo que era una recompensa en sí mismo? La idea era hacer que un ordenador fuera más utilizable, que fuera más interesante para los usuarios, que los ordenadores fueran tan interesantes que la gente se sintiera tentada a jugar con ellos, explorarlos y, finalmente, modificarlos. Cuando escribías un buen programa estabas construyendo una comunidad, no produciendo un producto. De todos modos, la gente no debería tener que pagar por el software: ¡la información debería ser gratuita!

• • • • • • • • •

Los hackers del TMRC no eran los únicos que habían estado ideando planes para el nuevo PDP-1. Durante ese verano de 1961, se estaba ideando un plan para el hackeo más elaborado hasta el momento: una muestra virtual de lo que podría resultar de una aplicación rigurosa de la Ética Hacker. El escenario de estas discusiones era un edificio de apartamentos en Higham Street en

Cambridge, y los perpetradores originales eran tres programadores itinerantes de veintitantos años que habían estado rondando varios centros de computación durante años. Dos de los tres vivían en el edificio, así que en honor a las pomposas proclamas que emanaban de la cercana Universidad de Harvard, el trío se refirió burlonamente al edificio como el Instituto Higham.

Uno de los miembros de esta institución falsa era Steve Russell, apodado, por razones desconocidas, Slug. Tenía ese patrón de habla de ardilla sin aliento tan común entre los hackers, junto con gafas gruesas, estatura modesta y un gusto fanático por las computadoras, las malas películas y la ciencia ficción pulp. Los tres intereses eran compartidos por los asistentes residentes a esas sesiones de toros en Higham Street.

Russell había sido durante mucho tiempo un "coolie" (para usar un término de TMRC) del tío John McCarthy. McCarthy había estado tratando de diseñar e implementar un lenguaje de alto nivel que pudiera ser suficiente para el trabajo de inteligencia artificial. Pensó que lo había encontrado en LISP. El lenguaje recibió su nombre por su método de procesamiento de listas; mediante comandos simples pero poderosos, LISP podía hacer muchas cosas con pocas líneas de código; también podía realizar poderosas recursiones (referencias a cosas dentro de sí mismo) que permitirían que los programas escritos en ese lenguaje realmente "aprendieran" de lo que sucedía mientras se ejecutaba el programa. El problema con LISP en ese momento era que ocupaba una cantidad enorme de espacio en una computadora, se ejecutaba muy lentamente y generaba cantidades voluminosas de código adicional a medida que se ejecutaban los programas, tanto que necesitaba su propio programa de "recolección de basura" para limpiar periódicamente la memoria de la computadora.

Russell estaba ayudando al tío John a escribir un intérprete de LISP para el enorme IBM 704. Era, en sus palabras, "un trabajo de ingeniería horrible", principalmente debido al tedioso procesamiento por lotes del 704.

Comparado con esa máquina, el PDP-1 parecía la Tierra Prometida para Slug Russell. ¡Más accesible que el TX-0 y sin procesamiento por lotes! Aunque no parecía lo suficientemente grande para hacer LISP, tenía otras capacidades maravillosas, algunas de las cuales fueron objeto de discusión en el Instituto Higham. Lo que particularmente intrigó a Russell y sus amigos fue la perspectiva de inventar algún tipo de elaborado "truco de pantalla" en el PDP-1, utilizando la pantalla CRT. Después de un largo discurso a medianoche, el Instituto Higham, compuesto por tres personas, se declaró en el acto insistiendo en que la demostración más eficaz de la magia del ordenador sería un juego visualmente impactante.

Se habían hecho varios intentos de hacer este tipo de cosas en el TX-0. Uno de ellos era un truco llamado Mouse in the Maze (Ratón en el laberinto): el usuario primero construía un laberinto con el lápiz óptico y un punto en la pantalla que representaba un ratón se abría paso tentativamente por el laberinto en busca de otro conjunto de puntos con forma de cuñas de queso. También había una “versión VIP” del juego, en la que el ratón buscaba copas de martini. Después de llegar al vaso, buscaba otro, hasta que se quedaba sin energía, demasiado borracho para continuar. Sin embargo, cuando se activaban los interruptores para hacer pasar al ratón por el laberinto una segunda vez, el ratón “recordaba” el camino hacia los vasos y, como un borracho experimentado, se apresuraba sin vacilar hacia la bebida. Eso era lo más lejos que llegaban los trucos de visualización en el TX-0.

Pero ya en el PDP-1, que tenía una pantalla más fácil de programar que la del TX-0, se habían producido algunos cambios importantes en la pantalla. El esfuerzo más admirado fue creado por uno de los gurús gemelos de la inteligencia artificial en el MIT, Marvin Minsky (el otro era, por

supuesto, McCarthy). Minsky era más extrovertido que su colega gurú de la IA y estaba más dispuesto a aceptar las necesidades de los demás. Minsky se había metido en el mundo de los hackers. Era un hombre con grandes ideas sobre el futuro de la informática: creía de verdad que un día las máquinas serían capaces de pensar y a menudo creaba un gran revuelo al llamar públicamente a los cerebros humanos «máquinas de carne», dando a entender que algún día las máquinas que no estuvieran hechas de carne también lo harían. Minsky, un hombre elfo de ojos brillantes detrás de unas gafas gruesas, una cabeza completamente calva y un omnipresente jersey de cuello alto, decía esto con su habitual estilo seco, orientado simultáneamente a maximizar la provocación y dejar solo una pista de que todo era una tontería cósmica (por supuesto, las máquinas no pueden pensar, je, je). Marvin era el auténtico; los hackers del PDP-1 solían asistir a su curso, Introducción a la IA 6.544, porque Minsky no solo era un buen teórico, sino que sabía de lo que hablaba. A principios de los años 60, Minsky estaba empezando a organizar lo que llegaría a ser el primer laboratorio del mundo en inteligencia artificial; Minsky sabía que para hacer lo que quería, necesitaría genios de la programación como soldados rasos, así que alentó el hackerismo de todas las formas que pudo.

Una de las contribuciones de Minsky al creciente canon de hacks interesantes fue un programa de visualización en el PDP-1 llamado Algoritmo del Círculo. En realidad, fue descubierto por error: mientras intentaba extraer una instrucción de un programa corto para convertir líneas rectas en curvas o espirales, Minsky confundió inadvertidamente un carácter “Y” con un “Y primo”, y en lugar de que la pantalla se garabateara en espirales incipientes como se esperaba, dibujó un círculo: un descubrimiento increíble, que más tarde se descubrió que tenía profundas implicaciones matemáticas. Para llevar el hackeo más allá, Minsky utilizó el Algoritmo del Círculo como punto de partida para una visualización más elaborada en la que tres partículas se influyeron entre sí y formaron fascinantes patrones en espiral en la pantalla, rosas autogeneradas con un número variable de hojas. “Las fuerzas que ejercían las partículas sobre otras eran totalmente extravagantes”, recordó Bob Wagner más tarde. “¡Estabas simulando una violación de la ley natural!”. Minsky llamó al hack un programa "Tri-Pos: Three-Position Display" (Visualización de tres posiciones), pero los hackers lo rebautizaron cariñosamente como Minskytron.

Slug Russell se inspiró en esto. En las sesiones del Instituto Higham algunos meses atrás, él y sus amigos habían discutido los criterios para el hack definitivo de visualización. Como habían sido fanáticos de la ciencia ficción basura, particularmente las novelas de ópera espacial de E.E. "Doc" Smith, de alguna manera decidieron que el PDP-1 sería una máquina perfecta para hacer una combinación de película de clase B y juguete de 120.000 dólares. Un juego en el que dos personas podrían enfrentarse en un enfrentamiento en el espacio exterior. Se organizó debidamente un Grupo de Estudio sobre Guerra Espacial del Instituto Higham, y su conclusión implicó fuertemente que Slug Russell debería ser el autor de este hack histórico.

Pero meses después, Russell ni siquiera había comenzado. Observaba al Minskytron crear patrones, activaba interruptores para ver cómo se desarrollaban nuevos patrones y, de vez en cuando, activaba más interruptores cuando el programa se quedaba atascado en la inactividad. Estaba fascinado, pero pensaba que el truco era demasiado abstracto y matemático. “Esta demo es una tontería”, decidió finalmente: sólo treinta y dos instrucciones y en realidad no hacía nada.

Slug Russell sabía que su juego de guerra en el espacio exterior haría algo. En sus propios términos kitsch y de ciencia ficción, sería absorbente de una manera en que ningún truco anterior lo había sido nunca. Lo que llevó a Slug a los ordenadores en primer lugar fue la sensación de poder que se

obtenía al ejecutar esas malditas cosas. Puedes decirle al ordenador lo que tiene que hacer y él lucha contigo, pero al final hace lo que le dices. Por supuesto, reflejará tu propia estupidez y, a menudo, lo que le dices que haga dará como resultado algo desagradable. Pero al final, después de torturas y tribulaciones, hará exactamente lo que quieres. La sensación que obtienes entonces es diferente a cualquier otra sensación en el mundo. Puede convertirte en un adicto. Eso convirtió a Slug Russell en un drogadicto, y pudo ver que había hecho lo mismo con los hackers que rondaban la Sala Kluge hasta el amanecer. Fue esa sensación la que lo provocó, y Slug Russell supuso que la sensación era poder.

Slug obtuvo una sensación similar, aunque menos intensa, de las novelas de Doc Smith. Dejó que su imaginación construyera la emoción de rugir a través del espacio en una nave espacial blanca... y se preguntó si esa misma emoción podría ser capturada mientras estaba sentado detrás de la consola del PDP-1. Esa sería la Guerra Espacial con la que soñaba. Una vez más, juró hacerlo.

Más tarde.

Slug no era tan ambicioso como algunos de los otros hackers. A veces necesitaba un empujón. Después de que cometió el error de abrir su bocaza sobre este programa que iba a escribir, los hackers del PDP-1, siempre ansiosos por ver otro hack agregado a la creciente pila de cintas de papel en el cajón, lo instaron a hacerlo. Después de murmurar excusas durante un rato, dijo que lo haría, pero que primero tendría que averiguar cómo escribir las elaboradas rutinas de seno-coseno necesarias para trazar el movimiento de las naves.

Kotok sabía que ese obstáculo podía resolverse fácilmente. En ese momento, Kotok se estaba volviendo bastante amigable con la gente de DEC, a varios kilómetros de distancia, en Maynard, Kotok se mostró muy cómodo con la gente. En DEC, los fabricantes de ordenadores eran informales y no consideraban a los hackers del MIT como los sucios y frívolos aficionados a los ordenadores que IBM podría haber tomado como ejemplos. Por ejemplo, un día, cuando se rompió un equipo, Kotok llamó a Maynard y se lo contó a DEC; le dijeron: «Venid y conseguid un repuesto». Cuando Kotok llegó, ya eran más de las cinco de la tarde y el lugar estaba cerrado. Pero el vigilante nocturno le dejó entrar, buscar el escritorio del ingeniero con el que había estado hablando y hurgar en él hasta encontrar la pieza. Informal, como les gusta a los hackers. Así que a Kotok no le supuso ningún problema ir un día a Maynard, donde estaba seguro de que alguien tendría una rutina para el seno y el coseno que pudiera funcionar en el PDP-1. Efectivamente, alguien lo tenía, y como la información era gratuita, Kotok se lo llevó al Edificio 26.

“Aquí tienes, Russell”, dijo Kotok, con las cintas de papel en la mano. “¿Y ahora cuál es tu excusa?”.

En ese momento, Russell no tenía excusa. Así que pasó sus horas libres escribiendo este juego de fantasía PDP-1, como nadie había visto antes. Pronto pasó sus horas “activas” trabajando en el juego. Empezó a principios de diciembre, y cuando llegó la Navidad, todavía estaba hackeando. Cuando el calendario llegó a 1962, todavía estaba hackeando. Para entonces, Russell podía producir un punto en la pantalla que se podía manipular: al pulsar algunos de los pequeños interruptores del panel de control, se podían hacer que los puntos aceleraran y cambiaran de dirección.

Luego se puso a hacer las formas de los dos cohetes: ambos eran cohetes clásicos de dibujos animados, puntiagudos en la parte superior y bendecidos con un par de aletas en la parte inferior. Para distinguirlos, hizo uno rechoncho y con forma de cigarro, con un bulto en el medio, mientras que al segundo le dio la forma de un tubo delgado. Russell utilizó las rutinas de seno y coseno para

averiguar cómo mover esas formas en diferentes direcciones. Luego escribió una subrutina para disparar un “torpedo” (un punto) desde la nariz del cohete con un interruptor en la computadora. La computadora escanearía la posición del torpedo y del barco enemigo; si ambos ocupaban la misma área, el programa llamaría a una subrutina que reemplazaría al barco descontento con una salpicadura aleatoria de puntos que representaban una explosión. (Ese proceso se llamó “detección de colisiones”). Todo esto fue en realidad un paso conceptual significativo hacia una programación “en tiempo real” más sofisticada, donde lo que sucede en una computadora coincide con el marco de referencia en el que los seres humanos están trabajando realmente. En otro sentido, Russell estaba emulando el estilo de depuración interactivo en línea que los hackers defendían: la libertad de ver en qué instrucción se detenía el programa y de usar interruptores o el Flexowriter para introducir una instrucción diferente, todo mientras el programa se ejecutaba junto con el depurador DDT. El juego Spacewar, un programa de computadora en sí mismo, ayudó a mostrar cómo todos los juegos (y tal vez todo lo demás) funcionaban como programas de computadora. Cuando te desviabas un poco, modificabas los parámetros y lo arreglabas. Ponías nuevas instrucciones. El mismo principio se aplicaba al tiro al blanco, la estrategia del ajedrez y los cursos del MIT. La programación de computadoras no era simplemente una actividad técnica, sino un enfoque a los problemas de la vida.

En las últimas etapas de la programación, Saunders ayudó a Slug Russell y hackearon algunas sesiones intensas de seis a ocho horas. En algún momento de febrero, Russell presentó el juego básico. Había dos naves, cada una con treinta y un torpedos. Había algunos puntos aleatorios en la pantalla que representaban estrellas en este campo de batalla celestial. Podías maniobrar las naves pulsando cuatro interruptores en la consola del PDP-1, que representaban el giro en el sentido de las agujas del reloj, el giro en el sentido contrario, la aceleración y el disparo de un torpedo.

Slug Russell sabía que al mostrar una versión preliminar del juego y dejar caer una cinta de papel con el programa en la caja con los programas del sistema del PDP-1, estaba dando la bienvenida a mejoras no solicitadas. Spacewar no era una simulación informática común: en realidad podías ser un piloto de cohete. Era Doc Smith cobrando vida. Pero el mismo poder que Russell había utilizado para crear su programa (el poder que el PDP-1 le prestaba a un programador para crear su propio pequeño universo) también estaba disponible para otros hackers, que naturalmente se sintieron libres de mejorar el universo de Slug Russell. Lo hicieron al instante.

La naturaleza de las mejoras podría resumirse en la reacción general de los hackers a la rutina original que Slug Russell utilizó para sus torpedos. Sabiendo que las armas militares en la vida real no siempre son perfectas, Russell pensó que haría que los torpedos fueran realistas. En lugar de hacerlos avanzar en línea recta hasta que se quedaran sin vapor y explotaran, introdujo algunas variaciones aleatorias en la dirección y la velocidad. En lugar de apreciar esta verosimilitud, los hackers la denunciaron. Les encantaban los sistemas que funcionaban sin problemas y las herramientas fiables, por lo que el hecho de que se quedaran atascados con algo que no funcionaba bien los volvía locos. Más tarde, Russell se dio cuenta de que “Las armas o herramientas que no son muy fiables se tienen en muy baja estima; a la gente le gusta poder confiar en sus herramientas y armas. Eso quedó muy claro en ese caso”.

Pero, por supuesto, eso se podía arreglar fácilmente. La ventaja que tenía un mundo creado por un programa informático sobre el mundo real era que se podía arreglar un problema grave como los torpedos defectuosos simplemente cambiando unas pocas instrucciones. ¡Por eso a tanta gente le resultó fácil perderse en el hackerismo en primer lugar! Así que se arreglaron los torpedos y la gente

pasó horas en duelos en el espacio exterior. Y aún más horas tratando de hacer que el mundo de Spacewar fuera mejor.

A Peter Samson, por ejemplo, le encantaba la idea de Spacewar, pero no podía soportar los puntos generados aleatoriamente que se hacían pasar por el cielo. El espacio real tenía estrellas en lugares específicos. “Tendremos lo real”, prometió Samson. Samson obtuvo un grueso atlas del universo y se puso a introducir datos en una rutina que escribió para generar las constelaciones reales visibles para alguien que estuviera en el ecuador en una noche clara. Todas las estrellas hasta la quinta magnitud estaban representadas; Samson duplicó su brillo relativo controlando la frecuencia con la que el ordenador iluminaba el punto de la pantalla que representaba la estrella. También manipuló el programa para que, a medida que avanzaba el juego, el cielo se desplazara majestuosamente: en cualquier momento la pantalla mostraba el cuarenta y cinco por ciento del cielo. Además de añadir verosimilitud, este programa de “Planetario caro” también proporcionó a los cazas cohete un fondo cartografiable desde el que medir la posición. El juego podría llamarse, como dijo Samson, Tiroteo en El Casiopea.

Otro programador, llamado Dan Edwards, no estaba satisfecho con el movimiento sin anclas de las dos naves en duelo. Convirtió el juego en una mera prueba de habilidades motoras. Pensó que añadir un factor de gravedad daría al juego un componente estratégico. Así que programó una estrella central —un sol— en el centro de la pantalla; uno podía usar la atracción gravitatoria del sol para obtener velocidad mientras giraba alrededor de él, pero si no tenía cuidado y se acercaba demasiado, sería arrastrado hacia el sol, lo que significaba una muerte segura.

Antes de que se pudieran emplear todas las implicaciones estratégicas de esta variación, Shag Garetz, uno de los tres integrantes del Instituto Higham, contribuyó con una característica de tipo impredecible. Había leído en las novelas de Doc Smith cómo los vehículos espaciales podían salir de una galaxia y entrar en otra gracias a un «tubo hiperespacial», que lo arrojaría a uno a «ese enigmático espacio N». Así que añadió una función de “hiperespacio” al juego, que permitía al jugador evitar una situación desesperada pulsando un botón de pánico que le llevaría a ese hiperespacio. Se permitía entrar al hiperespacio tres veces en el transcurso de una partida; el inconveniente era que nunca se sabía por dónde se podía salir. A veces reaparecías justo al lado del sol, justo a tiempo para ver cómo tu nave era arrastrada sin remedio a una muerte prematura en la superficie del sol. En homenaje al truco original de Marvin Minsky, Garetz programó la función de hiperespacio para que una nave que entrara en el hiperespacio dejara una “firma de emisión de estrés fotónico inducido por deformación”, una mancha de luz residual en una forma que a menudo se formaba tras una exhibición de Minskytron.

Las variaciones eran infinitas. Cambiando algunos parámetros se podía convertir el juego en una “guerra espacial hidráulica”, en la que los torpedos salían en corrientes eyaculadoras en lugar de uno a uno. O, a medida que la noche se hacía más tarde y la gente se quedaba atrapada en el modo interestelar, alguien podía gritar: “¡Enciendamos los Vientos del Espacio!” y alguien hackearía un factor de distorsión, que obligaría a los jugadores a hacer ajustes cada vez que se movieran. Aunque cualquier mejora que un hacker quisiera hacer sería bienvenida, era de muy mala educación hacer algún cambio extraño en el juego sin previo aviso. Las presiones sociales efectivas que imponía la Ética Hacker, que instaba a la práctica para mejorar, no para dañar, impedían cualquier instancia de ese tipo de travesuras. De todos modos, los hackers ya estaban involucrados en un ajuste alucinante del sistema: ¡estaban usando una computadora cara para jugar al juego más glorificado del mundo!

Spacewar se jugó muchísimo. Para algunos, era adictivo. Aunque nadie podía registrar oficialmente el PDP-1 para una sesión de Spacewar, cada momento libre de la máquina esa primavera parecía tener alguna versión del juego en funcionamiento. Botellas de Coca-Cola en la mano (y a veces con dinero en juego), los hackers organizaban torneos maratonianos. Russell finalmente escribió una subrutina que llevaría la puntuación, mostrándola en octal (todos podían Por aquel entonces, ya no se leía a primera vista ese sistema numérico de base ocho) el total de juegos ganados. Durante un tiempo, el principal inconveniente parecía ser que manejar los interruptores de la consola del PDP-1 era incómodo: a todos les dolían los codos por mantener los brazos en ese ángulo particular. Así que un día, Kotok y Saunders fueron al salón del club de TMRC y buscaron piezas para lo que se convertiría en los primeros joysticks de ordenador. Construidos totalmente con piezas que estaban por ahí, armaron las cajas de control, construyéndolas en una hora de inspiración y que fueron hechas de madera y tenían tapas de masonita, tenían interruptores para la rotación y el empuje, así como un botón para el hiperespacio. Por supuesto, todos los controles eran silenciosos, de modo que podías rodear subrepticiamente a tu oponente o agacharte en el espacio N, si así lo deseabas.

Mientras que algunos hackers perdieron el interés en Spacewar una vez que se calmó la furia de la fase de programación, otros desarrollaron un instinto asesino para idear estrategias para aniquilar a los oponentes. La mayoría de los juegos se ganaban y se perdían en los primeros segundos. Wagner se volvió experto en la estrategia de "esperar", en la que te quedabas en silencio mientras la gravedad te hacía girar alrededor del sol, para luego enderezarte y comenzar a disparar torpedos a tu oponente. Luego hubo una variación llamada "Apertura CBS", donde te inclinabas para disparar y luego girabas alrededor de la estrella: la estrategia recibió su nombre porque cuando ambos gladiadores de Spacewar la intentaban, dejaban un patrón en la pantalla que tenía un parecido notable con el ojo de CBS. Saunders, que se tomaba en serio su Spacewar, utilizó una estrategia CBS modificada para mantener el dominio a lo largo de los torneos; hubo un momento en el que no podía ser derrotado. Sin embargo, después de veinte minutos de proteger su lugar en el concurso estructurado al rey de la colina, incluso un maestro de Spacewarrior se ponía un poco borroso y más lento al sacar, y casi todos tenían la oportunidad de jugar Spacewar más de lo que probablemente era sensato. Peter Samson, segundo después de Saunders en Spacewarring, se dio cuenta de esto una noche cuando regresó a su casa en Lowell. Cuando salió del tren, miró hacia arriba al cielo nítido y claro. Un meteoro voló por encima. ¿Dónde está la nave espacial?, pensó Samson mientras giraba instantáneamente hacia atrás y agarraba el aire para un control. En mayo de 1962, durante la jornada de puertas abiertas anual del MIT, los hackers introdujeron en la máquina la cinta de papel con veintisiete páginas de código en lenguaje ensamblador PDP-1, instalaron una pantalla adicional (en realidad, un osciloscopio gigante) y ejecutaron Spacewar durante todo el día para un público que se desvió hacia allí y no podía creer lo que veía. La imagen (un juego de ciencia ficción escrito por estudiantes y controlado por un ordenador) estaba tan al borde de la fantasía que nadie se atrevió a predecir que con el tiempo se generaría un género entero de entretenimiento a partir de él.

No fue hasta años después, cuando Slug Russell estaba en la Universidad de Stanford, que se dio cuenta de que el juego era todo menos una aberración hacker. Después de trabajar hasta tarde una noche, Russell y algunos amigos fueron a un bar local que tenía algunas máquinas de pinball. Jugaron hasta la hora de cerrar; luego, en lugar de irse a casa, Russell y sus compañeros de trabajo volvieron a su ordenador, y lo primero que hicieron sus amigos fue ejecutar Spacewar. De repente, Russell se dio cuenta: “Esta gente dejó de jugar a la máquina de pinball y se puso a jugar a Spacewar; por Dios, es una máquina de pinball”. La máquina de pinball más avanzada, imaginativa y cara que el mundo haya visto.

Al igual que los ensambladores de los hackers y el programa de música, Spacewar no se vendió. Como cualquier otro programa, se guardó en un cajón para que cualquiera pudiera acceder a él, mirarlo y reescribirlo como quisiera. El esfuerzo colectivo que había mejorado el programa etapa por etapa podría haber servido como argumento a favor de la ética hacker: el afán de meterse en el funcionamiento de la cosa y mejorarla había llevado a una mejora mensurable. Y, por supuesto, todo fue muy divertido. No es de extrañar que otros propietarios de PDP-1 empezaran a oír hablar de ello, y las cintas de papel que contenían Spacewar se distribuyeran gratuitamente. En un momento dado, a Slug Russell se le ocurrió que tal vez alguien debería ganar dinero con esto, pero para entonces ya circulaban docenas de copias. DEC estaba encantado de recibir una copia, y los ingenieros de allí la usaron como un programa de diagnóstico final en las PDP-1 antes de que las lanzaran al mercado. Luego, sin borrar la memoria de la computadora, apagaban la máquina. El equipo de ventas de DEC lo sabía y, a menudo, cuando se entregaban máquinas a nuevos clientes, el vendedor encendía la energía, verificaba que no saliera humo por la parte posterior y seleccionaba la ubicación "VY" donde se encontraba Spacewar. Y si la máquina había sido empaquetada y enviada con cuidado, la estrella pesada estaría en el centro, y el cohete con forma de cigarro y el cohete con forma de tubo estarían listos para la batalla cósmica. Un vuelo inaugural para una máquina mágica.

• • • • • • •

Spacewar, como se vio después, fue el legado duradero de los pioneros del hacking del MIT. En los dos años siguientes, muchos de los aficionados a las TX-0 y PDP-1 abandonaron el Instituto. Saunders aceptaría un trabajo en la industria de Santa Mónica (donde más tarde escribiría un Spacewar para la PDP-7 que usaba en el trabajo). Bob Wagner se fue a la Rand Corporation. Peter Deutsch fue a Berkeley para comenzar su primer año de universidad. Kotok aceptó un trabajo a tiempo parcial que se convirtió en un importante puesto de diseño en el DEC (aunque se las arregló para quedarse en TMRC y la PDP-1 durante años después). En un proceso que tendría un impacto considerable en la difusión del hackerismo al estilo del MIT fuera de Cambridge, John McCarthy abandonó el Instituto para iniciar un nuevo laboratorio de inteligencia artificial en la Costa Oeste, en la Universidad de Stanford. Slug Russell, el siempre amable escritor de LISP de McCarthy, se unió a él.

Pero las nuevas caras y una mayor actividad en el campo de la informática asegurarían que la cultura hacker en el MIT no sólo continuara, sino que prosperara y se desarrollara más que nunca. Las nuevas caras pertenecían a hackers increíblemente audaces destinados a la fama de leyenda viviente a través del boca a boca. Pero los avances que permitirían a estas personas ocupar su lugar en la vida del sueño hacker ya estaban en marcha, iniciados por personas cuyos nombres se harían conocidos por medios más convencionales: artículos académicos, premios académicos y, en algunos casos, notoriedad en la comunidad científica.

Estas personas fueron los planificadores. Entre ellos había científicos que ocasionalmente se dedicaban al hackeo (Jack Dennis, McCarthy, Minsky), pero que en última instancia estaban más absorbidos por los objetivos de la informática que adictos al proceso informático. Consideraban que las computadoras eran un medio para una vida mejor para la raza humana, pero no creían necesariamente que trabajar con una computadora fuera el elemento clave para mejorar esa vida.

Algunos de los planificadores imaginaban un día en que las computadoras con inteligencia artificial aliviarían las cargas mentales del hombre, de la misma manera que la maquinaria industrial ya había

levantado parcialmente su yugo físico. McCarthy y Minsky eran la vanguardia de esta escuela de pensamiento, y ambos habían participado en una conferencia de Dartmouth en 1956 que estableció una base para la investigación en este campo. El trabajo de McCarthy en el lenguaje de alto nivel LISP estaba dirigido hacia este fin, y fue lo suficientemente intrigante como para animar a hackers como Slug Russell, Peter Deutsch, Peter Samson y otros a trabajar con LISP. Minsky parecía interesado en la inteligencia artificial con una base más teórica: un alegre Johnny Appleseed calvo en el campo, esparcía sus semillas, cada una de ellas una idea capaz de florecer en un verdadero manzano de técnicas y proyectos de IA útiles.

Los planificadores también estaban sumamente preocupados por poner el poder de las computadoras en manos de más investigadores, científicos, estadísticos y estudiantes. Algunos planificadores trabajaron para hacer que las computadoras fueran más fáciles de usar; John Kemeny de Dartmouth demostró cómo se podía hacer esto escribiendo un lenguaje de computadora más fácil de usar llamado BASIC. Los programas escritos en BASIC se ejecutaban mucho más lento que el lenguaje ensamblador y ocupaban más espacio de memoria, pero no requerían el compromiso casi monástico que exigía el lenguaje de máquina. Los planificadores del MIT se concentraron en extender el acceso real a las computadoras a más personas. Había todo tipo de justificaciones para ello, y la menor de ellas era la escala de la economía proyectada, que era claramente preferible al sistema vigente en aquel momento, en el que incluso segundos de tiempo de ordenador eran bienes valiosos (aunque nadie lo hubiera notado en la PDP-1, que jugaba a la guerra espacial). Si más gente utilizaba ordenadores, surgirían más programadores y teóricos expertos, y la ciencia de la informática (sí, estos planificadores agresivos la llamaban ciencia) sólo podría beneficiarse de ese nuevo talento. Pero había algo más en juego en todo esto. Era algo que cualquier hacker podía entender: la creencia de que la informática, en sí misma, era positiva. John McCarthy ilustró esa creencia cuando dijo que el estado natural del hombre era estar conectado a un ordenador todo el tiempo. “Lo que el usuario quiere es un ordenador que pueda tener continuamente a su disposición durante largos períodos de tiempo”.

El hombre del futuro. Manos sobre un teclado, ojos sobre un CRT, en contacto con el conjunto de información y pensamiento que el mundo había ido almacenando desde el comienzo de la historia. Todo sería accesible para el Hombre Computacional.

Nada de esto ocurriría con el IBM 704, que procesaba los datos por lotes. Tampoco ocurriría con el TX-0 y el PDP-1, cuyos registros semanales se rellenaban por completo en cuestión de horas tras colgarlos en la pared. No, para ello era necesario que varias personas utilizaran el ordenador a la vez (la idea de que cada persona tuviera su propio ordenador era algo que sólo un hacker consideraría interesante). Este concepto multiusuario se denominaba tiempo compartido, y en 1960 los planificadores más importantes del MIT crearon el Grupo de Estudio de Ordenadores de Largo Alcance. Entre sus miembros había gente que había observado con diversión y asentimiento el ascenso del hacker del MIT, gente como Jack Dennis, Marvin Minsky y el tío John McCarthy. Sabían lo importante que era que la gente pudiese hacerse con esas cosas. Para ellos, no se trataba de si compartir el tiempo o no, sino de cómo hacerlo.

Los fabricantes de ordenadores, en particular IBM, no estaban entusiasmados. Estaba claro que el MIT tendría que hacerlo prácticamente por sí solo. (La firma de investigación de Bolt Beranek y Newman también estaba trabajando en el tiempo compartido). Finalmente, se iniciaron dos proyectos en el MIT: uno fue el esfuerzo en gran medida en solitario de Jack Dennis para escribir por sí solo el primer sistema de tiempo compartido para el PDP-1. El segundo fue emprendido por

un profesor llamado F.J. Corbató, quien buscaría ayuda del reticente gigante IBM para escribir un sistema para el 7090.

El Departamento de Defensa, especialmente a través de su Agencia de Proyectos de Investigación Avanzada (ARPA), había estado apoyando a las computadoras desde la guerra, consciente de sus futuras aplicaciones para uso militar. Así que a principios de los años sesenta, el MIT había obtenido una subvención de largo plazo para su proyecto de tiempo compartido, que se llamaría Proyecto MAC (las iniciales significaban dos cosas: Computación de Acceso Múltiple y Cognición Asistida por Máquinas). El Tío Sam desembolsaría tres millones de dólares al año. Dennis estaría a cargo. Marvin Minsky también tendría una gran presencia, particularmente en el uso de la tercera parte del dinero que no se destinaría al desarrollo de tiempo compartido, sino al campo aún efímero de la inteligencia artificial. Minsky estaba encantado, ya que el millón de dólares era diez veces su presupuesto anterior para IA, y se dio cuenta de que una buena parte de los dos tercios restantes también se destinarían a actividades de IA. Era una oportunidad para crear una instalación ideal, donde la gente pudiera planificar la realización del sueño hacker con máquinas sofisticadas, protegidas de la locura burocrática del mundo exterior. Mientras tanto, el sueño hacker sería vivido día a día por estudiantes devotos de la máquina.

Los planificadores sabían que necesitarían gente especial para el personal de este laboratorio. Marvin Minsky y Jack Dennis sabían que el entusiasmo de hackers brillantes era esencial para hacer realidad sus Grandes Ideas. Como Minsky dijo más tarde sobre su laboratorio: “En este entorno estaban sucediendo varias cosas. Había teorías de inteligencia artificial muy abstractas en las que la gente estaba trabajando y algunos de [los hackers] estaban interesados en ellas, la mayoría no. Pero estaba la cuestión de cómo hacer los programas que hacen estas cosas y cómo hacer que funcionen”.

Minsky estaba muy contento de resolver esa cuestión dejándosela a los hackers, la gente para la que “los ordenadores eran lo más interesante del mundo”. El tipo de gente que, por diversión, hackeaba algo aún más disparatado que Spacewar y luego, en lugar de jugarlo toda la noche (como a veces sucedía en la sala Kluge), hackeaba un poco más. En lugar de simulaciones espaciales, los hackers que hacían el trabajo sucio en el Proyecto MAC se ocuparían de sistemas más grandes: brazos robóticos, proyectos de visión, acertijos matemáticos y sistemas laberínticos de tiempo compartido que dejaban atónitos a la imaginación. Afortunadamente, las clases que ingresaron al MIT a principios de los sesenta iban a proporcionar algunos de los hackers más devotos y brillantes que jamás se hayan sentado delante de una consola. Y ninguno de ellos encajaba tan plenamente con el título de “hacker” como Richard Greenblatt.

Capítulo 4. Greenblatt y Gosper

Ricky Greenblatt era un hacker en potencia. Años después, cuando era conocido en todos los centros informáticos del país como el hacker arquetípico, cuando las historias de su concentración absoluta eran casi tan prolíficas como los millones de líneas de código en lenguaje ensamblador que había hackeado, alguien le preguntaría cómo empezó todo. Se retorcía en su silla, luciendo menos desaliñado que cuando era estudiante, cuando tenía cara de querubín, cabello oscuro y una expresión dolorosamente torpe; la cuestión, pensó, se reducía a si los hackers nacen o se hacen, y de ahí surgió uno de los famosos non sequiturs que llegaron a conocerse como Blatt-ismos: “Si los hackers nacen, entonces se harán, y si se convierten en eso, nacieron”.

Pero Greenblatt admitiría que era un hacker nato.

No es que su primer encuentro con el PDP-1 hubiera cambiado su vida. Estaba muy interesado, claro. Había sido la semana de afluencia de estudiantes de primer año en el MIT, y Ricky Greenblatt tenía algo de tiempo libre antes de abordar sus cursos, listo para la gloria académica. Visitó los lugares que más le interesaban: la estación de radio del campus WTBS (la del MIT era quizás la única estación de radio universitaria del país con un exceso de ingenieros de sonido estudiantes y escasez de disc jockeys), el Tech Model Railroad Club y la sala Kluge en el edificio 26, donde se encontraba el PDP-1.

Algunos hackers estaban jugando a Spacewar.

La regla general era jugar con todas las luces de la sala apagadas, de modo que la gente que se agolpaba alrededor de la consola tuviera sus caras extrañamente iluminadas por esta exhibición de naves espaciales y estrellas pesadas. Caras extasiadas iluminadas por el resplandor de la computadora. Ricky Greenblatt estaba impresionado. Observó los choques cósmicos durante un rato, luego fue a la puerta de al lado para ver el TX-0, con sus estantes de tubos y transistores, sus elegantes fuentes de alimentación, sus luces e interruptores. En el club de matemáticas de su instituto, en Columbia (Misuri), había visitado el ordenador de procesamiento por lotes de la universidad estatal y había visto una gigantesca máquina clasificadora de tarjetas en una compañía de seguros local. Pero nada parecido. Aun así, a pesar de estar impresionado con la emisora de radio, el Club de Ferrocarriles a Escala y, sobre todo, con los ordenadores, se propuso entrar en el cuadro de honor.

Esta virtud académica no podía durar. Greenblatt, más aún que cualquier estudiante normal del MIT, era un voluntario conscripto del Imperativo de la Acción Práctica. Su vida había cambiado irrevocablemente el día de 1954 en que su padre, al visitar al hijo con el que no había vivido desde un divorcio temprano, lo llevó a la Unión de Estudiantes Memorial de la Universidad de Misuri, no lejos de la casa de Ricky en Columbia. Ricky Greenblatt se encariñó con el lugar de inmediato. No fue sólo por el cómodo salón, el televisor, el bar de refrescos... Fue por los estudiantes, que eran más a la altura intelectual de Ricky Greenblatt, de nueve años, que sus compañeros de clase. Iba allí a jugar al ajedrez y, por lo general, no tenía problemas para vencer a los estudiantes universitarios. Era un muy buen jugador de ajedrez.

Una de sus víctimas del ajedrez era un estudiante de ingeniería de la UM que recibía el subsidio para veteranos. Se llamaba Lester y el regalo de Lester a este niño prodigio de nueve años fue una introducción práctica al mundo de la electrónica. Un mundo en el que no había ambigüedades. La

lógica prevalecía. Tenías un cierto grado de control sobre las cosas. Podías construir cosas según tu propio plan. Para un niño de nueve años cuya inteligencia podría haberlo hecho sentir incómodo con sus compañeros cronológicos, un niño afectado por una separación matrimonial que era típica de un mundo de relaciones humanas que escapaban a su control, la electrónica era el escape perfecto.

Lester y Ricky trabajaron en proyectos de radioaficionados. Desarmaban viejos televisores. Antes de terminar la universidad, Lester le presentó a Ricky a un tal Sr. Houghton, que dirigía una tienda de radio local, y esa se convirtió en un segundo hogar para el joven durante la escuela secundaria. Con un amigo de la escuela secundaria, Greenblatt construyó una gama de proyectos peliagudos. Amplificadores, moduladores, todo tipo de artilugios de tubo de vacío de aspecto maligno. Un osciloscopio. Radios de aficionados. Una cámara de televisión. ¡Una cámara de televisión! Parecía una buena idea, así que la construyeron. Y, por supuesto, cuando llegó el momento de elegir una universidad, Richard Greenblatt eligió el MIT. Ingresó en el otoño de 1962.

El trabajo del curso fue rígido durante su primer semestre, pero Greenblatt lo manejaba sin muchos problemas. Había desarrollado una relación con algunos ordenadores del campus. Había tenido suerte, consiguiendo el curso optativo llamado EE 641 (Introducción a la programación informática) y a menudo iba a las máquinas de tarjetas perforadas en EAM para hacer programas para el enorme gigante 7090. Además, su compañero de habitación, Mike Beeler, había estado tomando un curso de algo llamado nomografía. Los estudiantes que tomaban la clase tenían acceso práctico a un IBM 1620, ubicado en otro enclave más de esos sacerdotes descarriados cuyas mentes habían sido nubladas por la niebla ignorante que provenía del equipo de ventas de IBM. Greenblatt solía acompañar a Beeler a 1620, donde perforaba su baraja de cartas y permanecía en la cola, cuando llegaba el turno, se introducían las tarjetas en el lector y se obtenía una impresión instantánea de una impresora. “Era algo divertido para hacer por las noches”, recordaría Beeler más tarde. “Lo hacíamos como otros podrían ver un partido deportivo o salir a tomar una cerveza”. Era algo limitado pero gratificante. Hizo que Greenblatt quisiera más.

En la época navideña, finalmente se sintió lo suficientemente cómodo como para pasar el rato en el Model Railroad Club. Allí, rodeado de gente como Peter Samson, era natural ponerse en modo hacker. (Las computadoras tenían varios estados llamados “modos”, y los hackers a menudo usaban esa frase para describir las condiciones de la vida real). Samson había estado trabajando en un gran programa de horarios para las sesiones operativas del TMRC en el trazado gigante; debido al procesamiento de números requerido, Samson lo había hecho en FORTRAN en el 7090. Greenblatt decidió escribir el primer FORTRAN para el PDP-1. Nunca pudo explicar por qué decidió hacer esto, y es probable que nadie le haya preguntado. Era común que, si uno quería hacer una tarea en una máquina y la máquina no tenía el software para hacerlo, escribiera el software adecuado para que pudiera hacerlo. Éste fue un impulso que Greenblatt luego elevaría a una forma de arte.

Él también lo hizo. Escribió un programa que le permitía escribir en FORTRAN, tomando lo que escribía y compilando el código en lenguaje de máquina, así como transformando las respuestas en lenguaje de máquina de la computadora nuevamente en FORTRAN. Greenblatt hizo su compilador de FORTRAN en gran parte en su habitación, ya que tenía problemas para obtener suficiente acceso al PDP-1 para trabajar en línea. Además de eso, se involucró en el trabajo de un nuevo sistema de relés debajo del diseño en TMRC. Parece que el yeso en la habitación (que siempre estaba bastante sucio de todos modos, porque el personal de limpieza tenía prohibida oficialmente la entrada) se seguía cayendo, y parte de él llegaba a los contactos del sistema que Jack Dennis había ideado a

mediados de los años cincuenta. Además, había algo nuevo llamado relé de resorte de alambre que parecía mejor que el antiguo. Así que Greenblatt pasó mucho tiempo esa primavera haciendo eso. Junto con el hackeo del PDP-1.

Es curioso cómo suceden las cosas. Empiezas a trabajar concienzudamente como estudiante, llegas a la lista de honores del decano y luego descubres algo que pone las clases en su perspectiva adecuada: son totalmente irrelevantes para el tema en cuestión. El tema en cuestión era el hackeo, y parecía obvio -al menos, tan obvio que nadie en el TMRC o el PDP-1 parecía pensar que era siquiera un tema útil de discusión- que el hackeo era una actividad tan satisfactoria que uno podía hacer de ello una vida. Si bien un ordenador es muy complejo, no es ni de lejos tan complejo como las diversas idas y venidas e interrelaciones del zoológico humano; pero, a diferencia del estudio formal o informal de las ciencias sociales, el hackeo no sólo te proporcionaba una comprensión del sistema, sino también un control adictivo, junto con la ilusión de que el control total estaba a sólo unas pocas características de distancia. Naturalmente, uno se pone a construir aquellos aspectos del sistema que parecen más necesarios para que funcione dentro del sistema de la manera adecuada. Con la misma naturalidad, trabajar en este sistema mejorado le permite saber más cosas que deben hacerse. Entonces, alguien como Marvin Minsky podría aparecer y decir: "Aquí hay un brazo robótico. Voy a dejar este brazo robótico junto a la máquina". Inmediatamente, nada en el mundo es tan esencial como crear la interfaz adecuada entre la máquina y el brazo robótico, y poner el brazo robótico bajo su control, y encontrar una manera de crear un sistema donde el brazo robótico sepa qué diablos está haciendo. Entonces puede ver cómo su descendencia cobra vida. ¿Cómo puede algo tan artificial como una clase de ingeniería compararse con eso? Es probable que su profesor de ingeniería nunca haya hecho nada tan interesante como los problemas que usted resuelve todos los días en el PDP-1. ¿Quién tiene razón?

En el segundo año de Greenblatt, el panorama informático en torno al PDP-1 estaba cambiando considerablemente. Aunque algunos de los hackers originales del TX-0 se habían marchado, había nuevos talentos llegando, y la nueva y ambiciosa configuración, financiada por el benévolo Departamento de Defensa, se adaptaba muy bien a sus actividades de hacking. Había llegado un segundo PDP-1; su hogar era el nuevo edificio rectangular de nueve pisos en Main Street, un edificio de una monotonía abrumadora, sin protuberancias y con ventanas sin alféizar que parecían pintadas sobre su superficie blanquecina. El edificio se llamaba Tech Square, y entre los clientes del MIT y corporativos que se mudaban allí estaba Project MAC. El noveno piso de este edificio, donde estaban las computadoras, sería el hogar de una generación de hackers, y ninguno pasaría tanto tiempo allí como Greenblatt.

A Greenblatt le pagaban (salarios inferiores al mínimo) por hackear como empleado estudiante, al igual que a varios hackers que trabajaban en el sistema o estaban empezando a desarrollar algunos de los grandes programas que harían inteligencia artificial. Empezaron a notar que este estudiante de segundo año torpemente educado era una potencial superestrella del PDP-1.

Estaba produciendo una cantidad increíble de código, hackeaba tanto como podía o se sentaba con una pila de hojas impresas y las marcaba. Iba y venía del PDP-1 al TMRC, con la cabeza fantásticamente conectada con las estructuras del programa en el que estaba trabajando o el sistema de relés que había hackeado bajo el diseño del TMRC. Para mantener esa concentración durante un largo período de tiempo, vivía, como varios de sus compañeros, la jornada de treinta horas. Era propicia para el hackeo intenso, ya que tenías un bloque extenso de horas de vigilia para ponerte en marcha con un programa y, una vez que estabas realmente en marcha, las pequeñas molestias como

el sueño no tenían por qué molestarte. La idea era quemar durante treinta horas, llegar al agotamiento total, luego ir a casa y derrumbarte durante doce horas. Una alternativa sería derrumbarte allí mismo en el laboratorio. Un pequeño inconveniente de este tipo de horario era que te ponía en desacuerdo con las rutinas que todos los demás en el mundo usaban para hacer cosas como cumplir con las citas, comer e ir a clases. Los hackers podían adaptarse a esto: uno solía hacer preguntas como "¿En qué fase está Greenblatt?" y alguien que lo había visto recientemente decía: "Creo que ahora está en una fase nocturna, y debería estar alrededor de las nueve o así". Los profesores no se adaptaban tan fácilmente a esas fases, y Greenblatt "se saltaba" sus clases.

Fue puesto en libertad condicional académica y su madre vino a Massachusetts para hablar con el decano. Tuvo que dar algunas explicaciones. "Su madre estaba preocupada", diría más tarde su compañero de habitación Beeler. "Su idea era que él estaba allí para obtener un título. Pero las cosas que estaba haciendo en la computadora eran completamente de última generación, nadie las estaba haciendo todavía. Vio cosas adicionales por hacer. Era muy difícil entusiasmarse con las clases". Para Greenblatt, no era realmente importante que estuviera en peligro de suspender la universidad. Hackear era primordial: era lo que mejor hacía y lo que lo hacía más feliz.

Su peor momento llegó cuando estaba tan "fuera de fase" que se quedó dormido después de un examen final. Esto sólo aceleró su salida del cuerpo estudiantil del MIT. Suspensión probablemente no habría hecho ninguna diferencia en su vida si no hubiera sido por una regla que decía que no se podía ser un empleado estudiante cuando se era un estudiante exiliado. Así que Greenblatt fue a buscar trabajo, con la plena intención de conseguir un trabajo de programación durante el día que le permitiera pasar las noches en el lugar donde quería pasar su tiempo: el noveno piso de Tech Square. Hackeando. Y eso es exactamente lo que hizo.

• • • • • • •

Había un hacker igualmente impresionante que había dominado el PDP-1 de una manera diferente. Más verbal que Greenblatt, era más capaz de articular mejor su visión de cómo la computadora había cambiado su vida, y cómo podría cambiar todas las vidas de nosotros. Este estudiante se llamaba Bill Gosper. Había empezado en el MIT un año antes que Greenblatt, pero había tardado un poco más en convertirse en un asiduo del PDP-1. Gosper era delgado, con rasgos de pájaro cubiertos por unas gafas gruesas y una cabeza rebelde de pelo castaño y rizado. Pero incluso un breve encuentro con Gosper era suficiente para convencerte de que allí estaba alguien cuya brillantez ponía cosas como la apariencia física en su perspectiva propiamente trivial. Era un genio de las matemáticas. En realidad, fue la idea de hackear el mundo de las matemáticas, más que hackear sistemas, lo que atrajo a Gosper a la computadora, y él iba a servir como un oponente durante mucho tiempo para Greenblatt y las otras personas orientadas a los sistemas en la sociedad de brillantes soldados rasos que ahora se formaban alrededor del nuevo Proyecto MAC.

Gosper era de Pennsauken, Nueva Jersey, al otro lado del río de Filadelfia, y su experiencia previa al MIT con las computadoras, como la de Greenblatt, se limitaba a ver a gigantes descomunales operar desde detrás de un panel de vidrio. Gosper recordaba vívidamente haber visto a la Univac del Instituto Franklin de Filadelfia imprimir imágenes de Benjamin Franklin en su impresora de línea. Gosper no tenía idea de lo que estaba pasando, pero parecía muy divertido.

Él mismo probó esa diversión por primera vez en su segundo semestre en el MIT. Había tomado un curso del tío John McCarthy, abierto sólo para estudiantes de primer año que habían obtenido promedios de calificaciones asquerosamente altos el semestre anterior. El curso comenzaba con

FORTRAN, continuaba con el lenguaje de máquina de IBM y terminaba con el PDP-1. Los problemas no eran triviales, cosas como trazar rayos a través de sistemas ópticos con el 709, o trabajar rutinas con un nuevo intérprete de punto flotante para el PDP-1.

El desafío de la programación atraía a Gosper. Especialmente en el PDP-1, que, después de la tortura del procesamiento por lotes de IBM, podía actuar en ti como un elixir embriagador. O tener sexo por primera vez. Años después, Gosper todavía hablaba con entusiasmo de “la emoción de tener ese teclado en vivo debajo de ti y que esa máquina respondiera en milisegundos a lo que estabas haciendo…”.

Aun así, Gosper se mostraba tímido a la hora de continuar con el PDP-1 una vez finalizado el curso. Estaba involucrado en el departamento de matemáticas, donde la gente le decía que sería mejor que se alejara de las computadoras, porque lo convertirían en un oficinista. El eslogan no oficial del departamento de matemáticas era “el de la computadora”. Gosper descubrió que su argumento era: «La informática no existe, ¡es brujería!». ¡Pues bien, Gosper sería un brujo! Se inscribió en el curso de Minsky sobre inteligencia artificial. El trabajo era de nuevo sobre el PDP-1, y esta vez Gosper se vio envuelto en el hacking en sí. En algún momento de ese semestre, escribió un programa para representar funciones en la pantalla, su primer proyecto real, y una de las subrutinas contenía un código de programa tan elegante que se atrevió a mostrárselo a Alan Kotok. Para entonces, Kotok había alcanzado, pensó Gosper, «el estatus de dios», no sólo por sus hazañas en el PDP-1 y el TMRC, sino por el hecho bien conocido de que su trabajo en DEC incluía un papel principal en el diseño de un nuevo ordenador, una versión muy mejorada del PDP-1. Gosper se puso eufórico cuando Kotok no sólo miró su hack, sino que pensó que era lo suficientemente inteligente como para mostrárselo a otra persona. ¡Kotok realmente pensó que había hecho algo genial! Gosper se puso a trabajar en el PDP-1.

Su gran proyecto en ese curso era un intento de “resolver” el juego Peg Solitaire (o HI-Q), en el que se tiene un tablero con la forma de un signo más con treinta y tres agujeros. Todos los agujeros, menos uno, están ocupados por una ficha: se saltan las fichas una sobre otra, eliminando las que se saltan. La idea es terminar con una ficha en el centro. Cuando Gosper y dos compañeros de clase le propusieron a Minsky que resolvieran el problema en el PDP-1, Minsky dudó de que pudieran hacerlo, pero agradeció el intento. Gosper y sus amigos no sólo lo resolvieron: “Lo demolimos”, diría más tarde. Hackearon un programa que permitiría al PDP-1 resolver el juego en una hora y media.

Gosper admiraba la forma en que la computadora resolvía HI-Q porque su enfoque era “contraintuitivo”. Tenía un profundo respeto por los programas que utilizaban técnicas que a primera vista parecían improbables, pero que de hecho aprovechaban la profunda verdad matemática de la situación. La solución contraintuitiva surgió de la comprensión de las conexiones mágicas entre las cosas en el vasto mandala de relaciones numéricas en las que se basaba en última instancia el hackeo. Descubrir esas relaciones (crear nuevas matemáticas en la computadora) iba a ser la búsqueda de Gosper; y a medida que comenzó a pasar más tiempo cerca del PDP-1 y el TMRC, se hizo indispensable como el principal "hacker matemático", no tan interesado en los programas de sistemas, pero capaz de idear algoritmos sorprendentemente claros (¡no intuitivos!) que podrían ayudar a un hacker de sistemas a eliminar algunas instrucciones de una subrutina o a desatascar un atasco mental al intentar ejecutar un programa.

• • • • • • •

Gosper y Greenblatt representaban dos tipos de hackeo en torno al TMRC y el PDP-1: Greenblatt se centraba en la construcción de sistemas pragmáticos y Gosper en la exploración matemática. Cada uno respetaba las virtudes del otro y ambos participaban en proyectos, a menudo colaborativos, que explotaban sus mejores habilidades. Más que eso, ambos eran importantes contribuyentes a la cultura aún naciente que empezaba a florecer en su forma más plena en el noveno piso de Tech Square. Por diversas razones, sería en este invernadero tecnológico donde la cultura crecería con más exuberancia, llevando la ética hacker a su extremo.

La acción se alternaría entre varias escenas. La Sala Kluge, con el PDP-1 ahora funcionando con el sistema de tiempo compartido, que Jack Dennis había trabajado durante un año en escribir, todavía era una opción para algunos hackers nocturnos, y especialmente para Spacewarring. Pero cada vez más, los verdaderos hackers preferían el ordenador Project MAC. Se encontraba entre otras máquinas en el noveno piso de Tech Square, duramente iluminado y con muebles estériles, donde uno podía escapar del zumbido de los acondicionadores de aire que hacían funcionar los distintos ordenadores solo metiéndose en una de varias oficinas diminutas. Por último, estaba el TMRC, con su máquina de Coca-Cola que nunca se vaciaba, la caja de cambio de Saunders y la sala de herramientas de al lado, donde la gente se sentaba a todas horas de la noche y discutía sobre cuestiones que para un extraño serían desconcertantemente arcanas.

Estas discusiones eran el alma de la comunidad hacker. A veces, la gente literalmente se gritaba, insistiendo en un determinado tipo de esquema de codificación para un ensamblador, o un tipo específico de interfaz, o una característica particular de un lenguaje informático. Estas diferencias hacían que los hackers golpearan la pizarra o arrojaran tiza por la habitación. No era tanto una batalla de egos como un intento de averiguar qué era “lo correcto”. El término tenía un significado especial para los hackers. Lo correcto implicaba que para cualquier problema, ya fuera un dilema de programación, un desajuste de la interfaz de hardware o una cuestión de arquitectura de software, existía una solución que era simplemente… eso. El algoritmo perfecto. Habrías llegado justo al punto justo, y cualquiera con medio cerebro vería que se había trazado la línea recta entre dos puntos y que no tenía sentido intentar superarla. “Lo correcto”, explicaría Gosper más tarde, “significaba muy específicamente la solución única, correcta y elegante... lo que satisfacía todas las restricciones al mismo tiempo, lo que todos parececían creer que esistía para la mayoría de los problemas”.

Gosper y Greenblatt tenían opiniones firmes, pero por lo general Greenblatt se cansaba de la corrosiva interacción humana y se alejaba para implementar algo. Elegante o no. En su forma de pensar, las cosas tenían que hacerse. Y si nadie más las iba a hackear, él lo haría. Se sentaba con papel y lápiz, o tal vez en la consola del PDP-1, y gritaba su código. Los programas de Greenblatt eran robustos, lo que significa que su base era firme, con controles de errores integrados para evitar que todo fracasara como resultado de un solo error. Cuando Greenblatt terminaba un programa, estaba completamente depurado. Gosper pensaba que a Greenblatt le encantaba encontrar y corregir errores más que a nadie que hubiera conocido, y sospechaba que a veces escribía código con errores solo para poder corregirlo.

Gosper tenía un estilo de hackeo más público. Le gustaba trabajar con público y, a menudo, los hackers novatos se sentaban detrás de él frente a la consola para verlo escribir sus ingeniosos hacks, que a menudo estaban repletos de breves y concisos puntos matemáticos de interés. Sus mejores momentos eran los hacks de exhibición, en los que un algoritmo inusual evocaba una serie de pirotecnias de CRT cada vez más impredecibles. Gosper actuaba como guía turístico a medida que

avanzaba, a veces enfatizando que incluso los errores tipográficos podían presentar un fenómeno numérico interesante. Mantenía una fascinación constante por la forma en que un ordenador podía escupir algo inesperado y trataba las expresiones de la máquina con infinito respeto. A veces, el evento aparentemente más aleatorio podía atraerlo hacia una tangente fascinante sobre las implicaciones de este absurdo cuadrático o esa función trascendental. Cierta magia de subrutinas en un programa de Gosper ocasionalmente se convertía en un memorando académico, como el que comienza:

> Basándome en la teoría de que las fracciones continuas están infrautilizadas, probablemente debido a su desconocimiento, ofrezco la siguiente sesión de propaganda sobre los méritos relativos de las fracciones continuas frente a otras representaciones numéricas.

Los debates en la sala de herramientas no eran simples sesiones de charlas universitarias. Kotok solía estar allí, y era en esas sesiones donde se tomaban decisiones importantes sobre el ordenador que estaba diseñando para DEC, el PDP-6. Incluso en su etapa de diseño, este PDP-6 se consideraba lo mejor en TMRC. Kotok a veces llevaba a Gosper de vuelta al sur de Jersey para las vacaciones, hablando mientras conducía sobre cómo este nuevo ordenador tendría dieciséis registros independientes. (Un registro, o acumulador, es un lugar dentro de un ordenador donde se produce el cálculo real. Dieciséis de ellos darían a una máquina una versatilidad hasta entonces inaudita). Gosper se quedaba sin aliento. ¡Ésa será, pensó, la mejor computadora de la historia del mundo!

Cuando DEC construyó la PDP-6 y entregó el primer prototipo al Proyecto MAC, todos pudieron ver que, si bien la computadora tenía todas las características necesarias para los usuarios comerciales, en el fondo era una máquina para hackers. Tanto Kotok como su jefe, Gordon Bell, recordando sus días en TX-0, usaron la PDP-6 para demoler las limitaciones que los habían molestado en esa máquina. Además, Kotok había escuchado atentamente las sugerencias de la gente de TMRC, en particular Peter Samson, quien se atribuyó el mérito de los dieciséis registros. El conjunto de instrucciones tenía todo lo que necesitabas y la arquitectura general era simétricamente sólida. Se podía acceder a los dieciséis registros de tres maneras diferentes cada uno, y se podía hacerlo en combinaciones, para lograr mucho con una sola instrucción. La PDP-6 también usaba una "pila", que permitía mezclar y combinar sus subrutinas, programas y actividades con facilidad. Para los hackers, la introducción del PDP-6 y su conjunto de instrucciones de una belleza dolorosa significó que tenían un nuevo y poderoso vocabulario con el que expresar sentimientos que antes sólo podían transmitirse en los términos más extraños. Minsky puso a los hackers a trabajar escribiendo un nuevo software de sistemas para el PDP-6, una hermosa máquina de color azul marino con tres grandes gabinetes, un panel de control más aerodinámico que el One, filas de brillantes interruptores en voladizo y una matriz de luces parpadeantes. Pronto se adentraron en la psicología de esta nueva máquina tan profundamente como lo habían hecho en el PDP-1. Pero en el Six se podía ir más allá. Un día, en la sala de herramientas de TMRC, los hackers estaban jugando con diferentes formas de realizar rutinas de impresión decimal, pequeños programas para hacer que la computadora imprimiera en números arábigos. A alguien se le ocurrió la idea de probar algunas de las nuevas y llamativas instrucciones del PDP-6, las que utilizaban la pila. Casi nadie había integrado estas nuevas instrucciones en su código; Pero cuando el programa se puso en la pizarra usando una instrucción llamada Push-J, para sorpresa de todos, la rutina de impresión decimal completa, que normalmente sería una página de código, salió con solo seis instrucciones. Después

de eso, todos en TMRC estuvieron de acuerdo en que Push-J había sido sin duda lo correcto para poner en el PDP-6.

Las discusiones y argumentos de Taller de Herramientas a menudo se trasladaba la comida a la cena, y la cocina elegida era casi siempre comida china. Era barata, abundante y, lo mejor de todo, disponible a altas horas de la noche. (Una segunda opción pobre era el cercano restaurante grasiento de la calle principal de Cambridge, un antiguo vagón de tren con paneles de color marrón llamado F&T Diner, pero al que los hackers llamaban "La Muerte Roja"). La mayoría de las noches de los sábados, o espontáneamente los días de semana después de las 10 p. m., un grupo de hackers salía, a veces en el Chevy convertible azul de 1954 de Greenblatt, al Chinatown de Boston.

La comida china también era un sistema, y la curiosidad de los hackers se aplicaba a ese sistema con tanta asiduidad como a un nuevo compilador LISP. Samson había sido un aficionado desde su primera experiencia en una excursión de TMRC a Joy Fong's en Central Square, y a principios de los sesenta había aprendido suficientes caracteres chinos para leer menús y pedir platos poco conocidos. Gosper se dedicó a la cocina con un vigor aún mayor; Samson recorría Chinatown en busca de restaurantes que abrieran después de medianoche y una noche encontró un pequeño local en un sótano regentado por una pequeña familia. La comida era bastante aburrida, pero vio que algunos chinos comían platos de aspecto fantástico. Así que decidió llevar a Samson allí.

Volvieron cargados de diccionarios chinos y exigieron un menú chino. El chef, un tal Sr. Wong, accedió de mala gana y Gosper, Samson y los demás estudiaron el menú como si fuera un conjunto de instrucciones para una nueva máquina. Samson proporcionó las traducciones, que fueron positivamente reveladoras. Lo que se llamaba "Carne con tomate" en el menú en inglés tenía un significado literal de Berenjena bárbara con cerdo y vaca. "Wonton" tenía un equivalente chino de Cloud Gulp. ¡Había cosas increíbles por descubrir en este sistema! Así que después de decidir las cosas más interesantes para pedir ("¿Alita de hibisco? Mejor pídela, averigua de qué se trata"), llamaron al Sr. Wong, y él parloteó frenéticamente en desaprobación china de sus selecciones. Resultó que se resistía a servirles la comida al estilo chino, pensando que los estadounidenses no la soportarían. El Sr. Wong los había confundido con los típicos estadounidenses tímidos, ¡pero estos eran exploradores! Habían estado dentro de la máquina y habían vivido para contarlo (lo contarían en lenguaje ensamblador). El Sr. Wong cedió. Salió la mejor comida china que cualquiera de los hackers había comido hasta la fecha.

La gente de TMRC era tan experta en hackear comida china que eventualmente pudieron superar a los dueños del restaurante. En una excursión de hackers un Día de los Inocentes, Gosper tuvo antojo de un plato poco conocido llamado melón amargo. Era una forma de pimiento verde salpicada de verrugas, con un intenso sabor a quinina que evocaba náuseas en todos, excepto en aquellos que habían adquirido dolorosamente el sabor. Por razones que solo él conoce, Gosper decidió tomarlo con salsa agridulce, y anotó el pedido en chino. La hija del dueño salió riendo. “Me temo que cometiste un error: mi padre dice que aquí dice ‘melón amargo agridulce’”. Gosper lo tomó como un desafío. Además, le ofendió que la hija ni siquiera supiera leer chino; eso iba en contra de la lógica de un sistema eficiente de restaurantes chinos, una lógica que Gosper había llegado a respetar. Así que, aunque sabía que su pedido era una petición absurda, actuó indignado y le dijo a la hija: “Por supuesto que dice melón amargo agridulce; los estadounidenses siempre pedimos melón amargo agridulce el primero de abril”. Finalmente, el dueño en persona salió. “¡No puedes comer!”, gritó. “¡No tiene sabor! ¡No tiene sabor!”. Los hackers se aferraron a la petición y el dueño regresó a la cocina.

El melón amargo agridulce resultó ser tan horrible como el dueño prometió. La salsa en ese lugar era terriblemente potente, tanto que si inhalabas mientras te ponías un poco en la boca te ahogabas. Combinado con el habitualmente repugnante melón amargo, creaba una sustancia química que parecía rechinar los dientes, y ninguna cantidad de té o Coca-Cola podía diluir ese sabor. Para casi cualquier otro grupo de personas, la experiencia habría sido una pesadilla. Pero para los hackers todo era parte del sistema. No tenía sentido humano, pero tenía su lógica. Era lo correcto; por lo tanto, todos los años, el Día de los Inocentes, volvían al restaurante e insistían en que su aperitivo fuera melón amargo agridulce.

Era durante esas comidas cuando los hackers eran más sociables. Los restaurantes chinos ofrecían a los hackers un sistema culinario fascinante y un entorno físicamente predecible. Para hacerlo aún más cómodo, Gosper, uno de los varios hackers que despreciaban el humo en el aire y desdeñaban a los que fumaban, trajo consigo un pequeño ventilador a batería. El ventilador era algo improvisado por un hacker adolescente que rondaba el laboratorio de inteligencia artificial: parecía una pequeña bomba malvada y había sido construido utilizando un ventilador de refrigeración de un ordenador desechado. Gosper lo colocaba sobre la mesa para soplar suavemente el humo en la cara de los infractores. En una ocasión, en el Lucky Garden de Cambridge, un deportista brutal sentado en una mesa cercana se indignó cuando el pequeño ventilador redirigió el humo a su mesa. Miró a esos tipos sucios del MIT con su pequeño ventilador y exigió a los hackers que apagaran el aparato. “Está bien, si deja de fumar”, dijeron, y en ese momento el deportista cargó contra la mesa, tirando platos, derramando té por todas partes e incluso clavando sus palillos en las aspas del ventilador. Los hackers, que consideraban el combate físico una de las interfaces humanas más idiotas, observaron con asombro. El incidente terminó tan pronto como el deportista vio a un policía sentado al otro lado del restaurante.

Esa fue una excepción a lo que normalmente eran reuniones agradables. La conversación giraba en torno a varios problemas del hacking. A menudo, la gente llevaba consigo sus impresiones y durante las pausas de la conversación enterraban sus narices en las montañas de código ensamblador. En ocasiones, los hackers incluso discutían algunos eventos del “mundo real”, pero la ética hacker sería identificable en los términos de la discusión. Se reduciría a algún fallo en un sistema. O se podría considerar un evento interesante a la luz de la curiosidad natural de un hacker sobre cómo funcionan las cosas.

Un tema común fue el horrible reinado de IBM, el asquerosamente desnudo emperador del reino de la informática. Greenblatt podía “encenderse en llamas” –una discusión extensa y agitada– sobre los millones de dólares que se desperdiciaban en computadoras IBM. Greenblatt se iba a casa de vacaciones y veía que el departamento de ciencias de la Universidad de Missouri, que supuestamente no tenía dinero, estaba gastando cuatro millones de dólares al año en el cuidado y la alimentación de un IBM descomunal gigante que no era ni de lejos tan ingenioso como el PDP-6. Y hablando de cosas tremendamente sobrevaloradas, ¿qué hay del sistema de tiempo compartido de IBM en el MIT, con ese IBM 7094 ahí mismo en el noveno piso? ¡Qué desperdicio!

Esto podría continuar durante toda una comida. Sin embargo, es revelador señalar las cosas de las que los hackers no hablaron. No dedicaron mucho tiempo a discutir las implicaciones sociales y políticas de las computadoras en la sociedad (excepto quizás para mencionar cuán completamente equivocada e ingenua era la concepción popular de las computadoras). No hablaron de deportes. En general, se reservaron sus vidas emocionales y personales, en la medida en que tenían alguna, para

sí mismos. Y para un grupo de hombres sanos en edad universitaria, hubo notablemente poca discusión de un tema que comúnmente obsesiona a grupos de esa composición: las mujeres.

Aunque algunos hackers llevaban vidas sociales algo activas, las figuras clave en el hackeo de TMRC-PDP se habían encerrado en lo que se llamaría "modo soltero". Era fácil caer en eso: por un lado, muchos de los hackers eran solitarios para empezar, socialmente incómodos. Era la previsibilidad y controlabilidad de un sistema informático, en oposición a los problemas irremediablemente aleatorios en una relación humana, lo que hacía que el hackeo fuera particularmente atractivo. Pero un factor aún más importante era la impresión de los hackers de que la informática era mucho más importante que involucrarse en una relación romántica. Era una cuestión de prioridades.

El hackeo había reemplazado al sexo en sus vidas.

“La gente estaba tan interesada en los ordenadores y ese tipo de cosas que realmente no tenían tiempo [para las mujeres]”, reflexionaría Kotok más tarde. “Y a medida que se hacían mayores, todo el mundo tenía la idea de que un día una mujer vendría y te daría un golpe en la cabeza y te diría: ¡tú!”. Eso fue más o menos lo que le pasó a Kotok, aunque no hasta que tenía casi treinta años. Mientras tanto, los hackers actuaban como si el sexo no existiera. No se fijaban en una mujer guapa sentada en la mesa de al lado en el restaurante chino, porque “el concepto de mujer guapa no estaba en el vocabulario”, explicó más tarde el hacker David Silver. Cuando una mujer entraba en la vida de un hacker serio, podía haber alguna discusión: “¿Qué le ha pasado a fulano? El tipo se está desmoronando por completo…”. Pero, en general, ese tipo de cosas no se despreciaban tanto como se las ignoraba. No podías pensar en aquellos que podrían haber quedado en el camino, porque estabas involucrado en lo más importante del mundo: hackear. Hackear no era solo una obsesión y un placer lujurioso, era una misión. Hackeabas y vivías según la Ética Hacker, y sabías que cosas terriblemente ineficientes y derrochadoras como las mujeres consumían demasiados ciclos y ocupaban demasiado espacio en la memoria. “Las mujeres, incluso hoy, son consideradas extremadamente impredecibles”, señaló un hacker del PDP-6, casi dos décadas después. “¿Cómo puede un hacker tolerar un ser tan imperfecto?”.

Tal vez hubiera sido diferente si hubiera habido más mujeres en TMRC y en el noveno piso; las pocas que andaban por allí se emparejaban con hackers. (“Nos encontraron”, diría más tarde un hacker). No había demasiadas de estas mujeres, ya que los forasteros, hombres o mujeres, a menudo se sentían desanimados por el grupo: los hackers hablaban de forma extraña, tenían horarios extraños, comían comida rara y pasaban todo el tiempo pensando en computadoras.

Y formaban una cultura exclusivamente masculina. El triste hecho era que nunca hubo una mujer con calidad de estrella hacker. Nadie sabe por qué. Había mujeres programadoras y algunas de ellas eran buenas, pero ninguna parecía tomar el hacking como una vocación sagrada como lo hacían Greenblatt, Gosper y los demás. Incluso el importante sesgo cultural contra las mujeres que quieren dedicarse a la informática seria no explica la absoluta falta de mujeres hackers. “Las cuestiones culturales son fuertes, pero no tanto”, concluiría Gosper más tarde, atribuyendo el fenómeno a diferencias genéticas o de “hardware”.

En cualquier caso, rara vez había mujeres presentes en las excursiones a los restaurantes chinos o en las sesiones en la sala de herramientas al lado del TMRC. Así que, naturalmente, no había que tener la mejor apariencia. Greenblatt, tal vez, llevó esto al extremo. Trabajó en varios proyectos

gigantescos a mediados de los sesenta y a menudo se enfrascaba tanto en ellos que sus hábitos personales se convertían en un asunto de cierta preocupación para sus compañeros hackers.

Después de abandonar la escuela, Greenblatt había aceptado un trabajo en una empresa llamada Charles Adams Associates, que estaba en proceso de comprar y montar un PDP-1. Greenblatt trabajaba en sus oficinas cerca de la "Autopista Tecnológica" de Boston, en las afueras de la ciudad, durante el día y conducía treinta millas de regreso al MIT después del trabajo para pasar la noche hackeando. Al principio se mudó de los dormitorios a la YMCA de Cambridge, pero lo echaron porque no mantenía su habitación limpia. Después de su temporada en Adams, lo volvieron a contratar en el Laboratorio de IA y, aunque tenía una situación de vida estable (como huésped en una casa de Belmont propiedad de un dentista jubilado y su esposa), a menudo dormía en un catre en el noveno piso. Al parecer, la limpieza no era una prioridad, ya que abundaban los cuentos sobre su notable suciedad. (Más tarde, Greenblatt insistiría en que no era peor que algunos de los otros). Algunos hackers recuerdan que una de las cosas que el hackeo de Greenblatt impedía era bañarse con regularidad, y el resultado era un olor muy fuerte. La broma en el laboratorio de IA era que había una nueva medida olfativa científica llamada miliblatt. Uno o dos miliblatts eran extremadamente potentes, y un blatt completo era casi inconcebible. Para reducir los miliblatts, cuenta la historia, los hackers llevaron a Greenblatt a un lugar en el pasillo del Edificio 20 donde había una ducha de emergencia para casos de exposición accidental a sustancias químicas, y la dejaron correr.

Gosper a veces criticaba a Greenblatt por sus hábitos personales, y le molestaba especialmente la costumbre de Greenblatt de frotarse las manos, lo que hacía que cayeran pequeños trozos de tierra. Gosper los llamaba blattlies. Cuando Greenblatt trabajaba en el escritorio de Gosper y dejaba blattlies, Gosper se aseguraba de lavar el área con amoníaco. Gosper también se burlaba a veces de Greenblatt por sus extraños patrones de habla, su tos frecuente, su mala ortografía, sus murmullos, aunque muchas de las expresiones de Greenblatt se integraron en el vocabulario específico que todos los hackers usaban hasta cierto punto. Por ejemplo, probablemente fue Greenblatt quien popularizó la práctica de duplicar las palabras para enfatizar, como las veces que se ponía nervioso al explicar algo a Gosper, Kotok y Samson, y las palabras se enredaban, y él suspiraba, diciendo "Oh, la-la-la-la" y comenzaba de nuevo. Gosper y los demás se reían, pero, como una familia adopta los patrones de habla de un bebé y los graciosos impropropismos, la comunidad adoptó muchos de los Greenblattismos.

A pesar de estos extraños rasgos personales, los hackers sentían un gran respeto por Greenblatt. Era como era debido a prioridades conscientes: era un hacker, no un miembro de la alta sociedad, y no había nada más útil que hackear. Lo consumía tanto que a veces pasaba seis meses sin encontrar tiempo para cobrar su sueldo del MIT. “Si se hubiera sentado al azar y hubiera intentado articular lo que estaba pensando y haciendo todo el tiempo, no habría logrado hacer nada”, diría Gosper más tarde. “Si se hubiera preocupado por cómo se escriben las cosas, no habría logrado escribir nada. Hacía lo que se le daba bien. Era un pragmático absoluto. Lo que la gente pensara, al diablo. Si alguien pensaba que era estúpido o un empollón, ese era su problema. Algunas personas lo pensaban, y estaban equivocadas”.

Gosper podía apreciar la obstinación de Greenblatt porque su propia insistencia en graduarse (lo que hizo en 1965) lo había metido en problemas. No fue que su último año en el MIT fuera un desastre académico, porque logró cumplir con los requisitos de graduación por un estrecho margen. El problema era un pacto que había hecho con la Marina de los Estados Unidos. Antes de ingresar al

MIT, había presentado un examen de servicio civil y había obtenido una calificación lo suficientemente alta como para ser incluido en un programa exclusivo de desarrollo de ingeniería para estudiantes. Trabajó durante los veranos para la Marina, que pagaba la mitad de su matrícula y le exigía que trabajara allí durante tres años después de graduarse. Cuando Gosper se alistó, había una cláusula de escape que le permitía posponer su compromiso si iba a la escuela de posgrado; y si podía lograr que una corporación pagara la inversión de tres mil dólares de la Marina después de eso, ya no estaría obligado. Pero durante el último año de Gosper, la laguna jurídica de la escuela de posgrado se cerró. Sólo una indemnización podría salvarlo, y él no tenía el dinero.

La perspectiva de entrar en la Marina era espantosa. Durante sus empleos de verano había estado expuesto a un sistema patético que era la antítesis de la ética hacker. Los programadores eran mantenidos en una habitación totalmente separada de la máquina; a veces, como recompensa por años de servicio, dejaban que un trabajador particularmente obediente se aventurara a entrar en la sala de computadoras y viera su programa en funcionamiento. (Se cuenta que a una mujer se le permitió este privilegio, y la visión de las luces parpadeantes y los discos zumbando le hizo desmayarse.) Además, el jefe de Gosper en la Marina era un hombre que no podía entender por qué el logaritmo de las sumas en una ecuación dada no era la suma de los logaritmos. No había forma de que Bill Gosper fuera a trabajar para un hombre que no sabía por qué el logaritmo de la suma no era la suma de los logaritmos.

Gosper también tenía la impresión de que la Marina estaba en la cama con Univac. Consideraba que la máquina Univac era una parodia grotesca de un ordenador, un gigante descomunal. La Marina tenía que saber que era un ordenador básicamente falso, pensó, pero lo utilizaba de todos modos: era un ejemplo clásico del inevitable resultado deformado de la burocracia del Mundo Exterior. Vivir con esa máquina sería sumergirse en el infierno. Gosper utilizaba ordenadores para buscar cosas que nadie había encontrado antes, y era esencial que el ordenador que utilizase fuera óptimo en todos los sentidos. El PDP-6 era lo mejor que había encontrado hasta el momento, y estaba decidido a no dejarlo, especialmente por un perro como el Univac. «Si veo que una máquina tiene algún problema increíblemente estúpido, algún error en su diseño o lo que sea, me irrita muchísimo», explicaría Gosper más tarde. «Mientras que el PDP-6 siempre me pareció una máquina infinitamente perfectible. Si había algo mal, lo cambiabas. En cierto sentido, vivíamos dentro de la maldita máquina. Era parte de nuestro entorno. Había casi una sociedad allí... No podía imaginarme estar sin un PDP-6".

Gosper estaba decidido a encontrar el dinero para pagar a la Marina, y a ganarlo mientras trabajaba para una empresa con un PDP-6. Cumplió estos estrictos criterios al conseguir un trabajo en la empresa para la que Greenblatt había trabajado el año anterior, Charles Adams. El hecho de que la empresa Adams nunca consiguiera que su PDP-6 funcionara correctamente (Greenblatt insiste en que hizo su parte de la preparación adecuadamente) no pareció molestar a Gosper: lo que le asustó fue el hecho de que Charles Adams desechara el proyecto y comprara una copia al carbón del mismo Univac gigante descomunal que tenía la Marina.

Pero para entonces ya había llegado más financiación para el Proyecto MAC, y Bill Gosper encontró su camino hacia la nómina. No tuvo que cambiar sus hábitos, ya que durante toda su estancia en Adams había estado trabajando en el PDP-6 en el noveno piso todas las noches.

• • • • • • • •

Para entonces, Greenblatt estaba en plena actividad de hacking. Uno de los primeros proyectos en los que trabajó en el PDP-6 fue un compilador LISP, para permitir que la máquina ejecutara la última y más ingeniosa versión del lenguaje de inteligencia artificial de Jobn McCarthy. El joven Peter Deutsch había escrito un LISP para el PDP-1, pero no era demasiado efectivo, ya que el One tenía menos memoria; y LISP, que trabaja con símbolos y no con números fácilmente traducibles a binario, consume una increíble cantidad de memoria.

Algunas personas, en particular Gosper, pensaron que LISP también sería una pérdida de tiempo en el PDP-6. Gosper siempre se preocupó por lo que él consideraba la atroz falta de potencia informática en aquellos días, y más tarde se maravillaría de lo ignorantes que eran todos en el laboratorio de IA, intentando realizar tareas imposibles y culpando de sus fracasos no a las insignificantes máquinas que tenían, sino a ellos mismos. En su último año, Minsky le había puesto a Gosper a trabajar en una pantalla que comprobaría si un determinado fenómeno visual era binocular o monocular. Gosper se las arregló para acercarse con una ingeniosa forma de trébol, que al menos mostraba el fenómeno, pero en general se golpeaba la cabeza contra la pared tratando de que la máquina hiciera más de lo que podía hacer. Una de las tareas que Gosper consideraba imposible era un LISP útil en un PDP-6: podría ser bueno como evaluador de símbolos, pero no para hacer nada. Lo consideraba una de las locuras de Minsky que Greenblatt y los demás habían sido engañados para implementar.

Pero Greenblatt vio más. Aunque se dio cuenta de que el LISP en el PDP-6 sería hasta cierto punto un hack, no totalmente pragmático, vio la necesidad de avanzar hacia él. Era un lenguaje poderoso que ayudaría al campo de la inteligencia artificial a avanzar: era el lenguaje con el que las computadoras realizarían tareas extremadamente difíciles, con el que realmente podrían aprender. Greenblatt estaba empezando entonces a tener cierta visión del futuro, un indicio de una implementación técnica del sueño hacker. Así que él y algunos otros -incluso Kotok vino de DEC- comenzaron a implementar el LISP en el PDP-6. Llenaron las pizarras de TMRC con capas y capas de código, y finalmente lo conseguí poner en marcha en la máquina.

Las secciones cruciales fueron escritas por Greenblatt y otro hacker. Se consideraba que dos o tres personas en un proyecto eran lo correcto, mucho menos que el llamado estilo de “ola humana” de IBM, que arrojaba a docenas de programadores a un problema y terminaba con basura. Y era mejor confiar en dos o tres personas que en un solo cruzado, de modo que cuando una persona estaba al final de su fase de treinta horas, alguien más podía entrar y seguir hackeando. Una especie de proyecto de equipo.

Con PDP-6 MacLISP (llamado así por el Proyecto MAC), los hackers comenzaron a integrar ese lenguaje de computadora en sus programas, e incluso en su conversación. La convención LISP de usar la letra “p” como predicado, por ejemplo, fue la inspiración para un estilo común de los hackers de hacer una pregunta. Cuando alguien decía “¿Comida-P?”, cualquier hacker sabía que le estaban preguntando si quería comer algo. Los términos LISP “T” y “nil” pasaron a significar, respectivamente, “sí” y “no”. La aceptación de LISP no disminuyó el amor de los hackers por el lenguaje ensamblador, en particular el elegante conjunto de instrucciones PDP-6. Pero como Greenblatt e incluso Gosper se dieron cuenta más tarde, LISP era un poderoso constructor de sistemas que encajaba perfectamente en la ética práctica de los hackers.

DEC había mostrado interés en MacLISP, y Kotok hizo arreglos para que Greenblatt y los demás fueran a Maynard a altas horas de la noche para trabajar en el programa, escribir su código y

depurarlo. Todo era parte del sencillo acuerdo entre MIT y DEC, y nadie lo cuestionó. Lo correcto era asegurarse de que cualquier buen programa tuviera la mayor exposición posible, porque la información era gratuita y el mundo solo mejoraría con su flujo acelerado.

Después de trabajar en MacLISP, Greenblatt fue quizás el más autorizado de los hackers de sistemas del PDP-6. El nuevo administrador del laboratorio de IA, un joven del suroeste llamado Russell Noftsker, había contratado a Greenblatt principalmente para mantener y mejorar la creación orgánica que es un sistema operativo de computadora. Pero la visión de Greenblatt no se detenía en los sistemas; se sentía intensamente atraído por los conceptos de inteligencia artificial. Decidió utilizar el sistema para hacer algo en ese ámbito y, como había sido ajedrecista toda su vida, era lógico que trabajara en un programa de ajedrez que fuera mucho más allá del esfuerzo de Kotok y de los otros proyectos de ajedrez con IA que se habían intentado en varios laboratorios de todo el país.

Como cualquier buen hacker, tan pronto como decidía hacer algo, comenzaba a trabajar en ello. Nadie le pidió una propuesta. No se molestó en notificar a sus superiores. Minsky no tuvo que ponderar las virtudes relativas del proyecto. No había canales por los que pasar porque a mediados de los sesenta, en aquellos primeros días del laboratorio de IA, los propios hackers eran los canales. Fue la ética hacker puesta en práctica, y Greenblatt la aprovechó al máximo.

Había visto una partida jugada por el programa Kotok y pensaba que era una porquería. Básicamente, esos tipos no sabían jugar al ajedrez: influenciados por el romanticismo de una computadora que hacía movimientos, de alguna manera habían olvidado la idea de que el nombre del juego era tomar las piezas del otro tipo. El programa de Greenblatt usaba sofisticadas técnicas de inteligencia artificial para intentar averiguar movimientos de acuerdo con ciertos criterios que él consideraba buen ajedrez. Trabajando con un par de hackers más, Greenblatt se puso a codificar. Consiguió tener cuatro horas de tiempo de PDP-6 al día, y siguió escribiendo sin conexión cuando no estaba en la máquina. Consiguió que el programa jugara ajedrez de verdad en una semana. El programa fue depurado, se le dieron funciones y, en general, se mejoró durante los meses siguientes. (Finalmente, a Greenblatt le ofrecieron un título del MIT si escribía una tesis sobre su programa de ajedrez; nunca lo hizo.)

En 1965, en el MIT circulaba un famoso memorándum de la Rand Corporation titulado “Alquimia e inteligencia artificial”. Su autor, un académico llamado Herbert Dreyfus, arremetió contra el campo y sus practicantes. Para los hackers, su crítica fue particularmente nociva, ya que la computadora era su modelo implícito de comportamiento, al menos en sus teorías de la información, la justicia y la acción. Dreyfus se centró en la estructura ridículamente limitada de la computadora (en comparación con la estructura del cerebro humano). Su golpe de gracia fue la afirmación contundente de que ningún programa de computadora sería capaz de jugar una partida de ajedrez lo suficientemente buena como para vencer a un niño de diez años.

Después de que Greenblatt terminó su programa de ajedrez, llamado MacHack, el MIT invitó a Dreyfus a jugar con el PDP-6. Los hackers se reunieron para ver al sustituto informático de Richard Greenblatt jugar con este oponente anticomputadora engreído, delgado, pelirrojo y con gafas. El pionero de la inteligencia artificial Herbert Simon, que vio la partida, fue citado más tarde diciendo que fue

> ... un verdadero suspenso. Son dos empujadores de madera... luchando entre sí... Dreyfus estaba siendo derrotado bastante y luego encontró un movimiento que podría haber capturado la reina del oponente. Y la única manera en que el oponente podía salir de esta situación era mantener a Dreyfus en jaque con su propia reina hasta que pudiera hacer un tenedor con la reina y el rey e intercambiarlos. Y el programa procedió a hacer exactamente eso. Tan pronto como lo hizo, el juego de Dreyfus se vino abajo, y luego lo hizo jaque mate justo en el medio del tablero.

Peter Samson recordó más tarde la escena inmediatamente posterior a la derrota de Dreyfus: el crítico derrotado miró a los profesores y hackers del MIT reunidos, incluido un Greenblatt victorioso, con una mirada de perplejidad. ¿Por qué no estaban vitoreando, aplaudiendo, restregándoselo en la cara? Porque lo sabían. Dreyfus era parte de ese "mundo real" que no podía comprender la naturaleza asombrosa de las computadoras, o cómo era trabajar con computadoras tan de cerca que una PDP-6 podía realmente convertirse en su entorno. Esto era algo que Dreyfus nunca sabría. Incluso Minsky, que nunca se sumergió realmente en el baptisterio de lenguaje ensamblador, que funcionaba durante treinta horas al día y siete días a la semana, no había experimentado lo que habían experimentado los hackers. Los hackers, los Greenblatt y los Gosper, estaban seguros de haber estado allí, de saber cómo era y de haber regresado allí, de haber producido, de haber descubierto cosas, de haber hecho que su mundo fuera diferente y mejor. En cuanto a convencer a los escépticos, de hacer que el mundo exterior se enterara del secreto, de hacer proselitismo en favor de la ética hacker, todo eso no era ni de lejos tan interesante como vivirlo.

Capítulo 5. La Sociedad de Cableado Informático de Medianoche

Greenblatt era un hacker de sistemas y un visionario de las aplicaciones; Gosper era un explorador metafísico y un manitas de lo esotérico. Juntos eran dos patas de un triángulo tecnocultural que serviría como base de la Ética Hacker en su ascenso a la supremacía cultural en el MIT en los años venideros. La tercera pata del triángulo llegó en el otoño de 1963, y su nombre era Stewart Nelson.

No mucho después de su llegada, Stew Nelson mostró su curiosidad y habilidad para adentrarse en reinos electrónicos inexplorados, rasgos que indicaban su potencial para convertirse en un mago maestro al servicio de la Ética Hacker. Como era la costumbre, Nelson había llegado una semana antes para el Freshman Rush. Era un chico bajito, generalmente taciturno, con cabello rizado, ojos marrones penetrantes y una gran sobremordida, que le daba el aspecto inquieto y curioso de un pequeño roedor. De hecho, Stewart Nelson estaba buscando equipos electrónicos sofisticados con los que pudiera tocar, y no tardó mucho en encontrar lo que quería en el MIT.

Todo empezó en WTBS, la estación de radio del campus. Bob Clements, un estudiante que trabajaba en la estación y que más tarde se dedicaría a hackear PDP-6, estaba mostrando a un grupo de estudiantes de primer año las salas de control cuando abrió una puerta que daba a la compleja maquinaria y encontró a Stew Nelson, “un niño cobarde”, como recordaría más tarde, “que tenía sus dedos en las entrañas de nuestras líneas telefónicas y nuestro transmisor de radio del Campus Este”.

Al final, encontró el PDP-1 en la Sala Kluge. La máquina entusiasmó mucho a Stewart Nelson. Vio una computadora amigable que se podía tocar con las manos y, con una confianza que provenía de lo que Greenblatt podría llamar hackerismo nato, se puso a trabajar. Se dio cuenta de inmediato de que el altavoz externo del One estaba conectado a la computadora y de que el programa de música de Peter Samson podía controlar ese altavoz. Así que una noche, muy tarde, cuando John McKenzie y la gente que atendía el TX-0 de al lado dormían en sus casas, Stewart Nelson se puso a aprender a programar el PDP-1, y no le llevó mucho tiempo enseñarle algunos trucos nuevos. Había programado algunos tonos apropiados para que salieran del altavoz y entraran en el receptor abierto del teléfono del campus que se encontraba en la Sala Kluge. Estos tonos hicieron que el sistema telefónico se pusiera en posición de atención, por así decirlo, y bailara. ¡Bailen, líneas telefónicas, bailen!

Y las señales bailaron. Bailaron de un lugar en el sistema de líneas de enlace del MIT al siguiente y luego al Observatorio Haystack (conectado al sistema del MIT), donde bailaron al ritmo de una línea abierta y, así liberadas, bailaron hacia el mundo. No había forma de detenerlos, porque los tonos particulares que Stew Nelson había generado en el PDP-1 eran los tonos exactos que la compañía telefónica usaba para enviar sus llamadas internas a todo el mundo, y Stew Nelson sabía que le permitirían recorrer todo el maravilloso sistema que era la compañía telefónica, sin pagar un centavo.

Este alquimista analógico, el nuevo rey de los hackers, estaba mostrando a un grupo profundamente impresionado de programadores de PDP-1 cómo un estudiante universitario solitario de primer año podía arrebatar el control del sistema telefónico de casi cien años de antigüedad, utilizándolo no con fines de lucro sino para explorar por placer. Se corrió la voz de estas hazañas, y Nelson comenzó a alcanzar un estatus heroico en TMRC y en la Sala Kluge; pronto algunos de los miembros más aprensivos del PDP-1 se lamentaban de que no hubiera ido demasiado lejos. Greenblatt no lo creía, ni tampoco ningún hacker verdadero: la gente había hecho ese tipo de cosas en TMRC durante años; y si Nelson llevó las cosas un paso más allá, eso fue un resultado positivo de la Ética Hacker.

Pero cuando John McKenzie se enteró, le ordenó a Nelson que parara, probablemente porque se dio cuenta de que no había mucho que pudiera hacer para frenar la eterna búsqueda de conocimientos de sistemas de Stew Nelson. “¿Cómo se puede detener un talento como ese?”, reflexionó más tarde. Al final resultó que las cosas iban a ir mucho más allá antes de que Stewart Nelson terminara. En cierto modo, nunca se detendrían.

La pirotecnia de primer año de Nelson no era tan sorprendente a la luz de su vida antes del MIT. Nacido en el Bronx, Nelson era hijo de un físico convertido en ingeniero que había hecho un trabajo pionero en el diseño de televisores en color. Sin embargo, el propio interés de Stewart por la electrónica no necesitaba el estímulo de sus padres. Era tan natural como caminar, y a los cinco años ya construía radios de cristal. A los ocho, estaba trabajando en alarmas antirrobo de doble relé. Tenía poco interés, social o educativo, en la escuela, pero gravitó hacia la tienda de electrónica, donde se dedicaba a la experimentación incesante. No pasó mucho tiempo antes de que las madres de los otros niños prohibieran a sus hijos jugar con Stewart, temían que su progenie se freiría con una dosis de electricidad. Éstos eran los peligros inevitables de jugar con circuitos de válvulas de vacío potentes y transistores de última generación alimentados por líneas eléctricas de 110 V. En ocasiones, Stew recibía descargas tan severas que se sacudía dolorosamente. Más tarde contaría historias de su equipo volando a través de la habitación y estallando en pedazos. Después de una descarga particularmente abrasadora, juró no volver a jugar con electricidad. Pero después de unos dos días volvió a hacerlo, como un joven solitario que trabajaba en proyectos fantásticos.

A Stew le encantaba el teléfono. Su familia se había mudado a Haddonfield, Nueva Jersey, y pronto descubrió que haciendo clic en los interruptores en los que se apoya el receptor, en realidad se podía marcar un número. Alguien en el otro extremo diría: "Hola... ¿sí? "¿Hola?" y te das cuenta de que no se trata de un simple aparato cualquiera, sino de algo conectado a un sistema que puedes explorar sin fin. Stewart Nelson pronto estaba construyendo cosas que pocos de sus vecinos a mediados de los años 50 habían visto, como marcadores automáticos y aparatos que podían conectarse a varias líneas telefónicas, recibiendo una llamada en una línea y llamando automáticamente en la otra. Aprendió a manejar el equipo telefónico con la destreza con la que un artista maneja sus herramientas; los testigos contarían más tarde cómo Nelson, cuando se enfrentaba a un teléfono, lo desmantelaba inmediatamente, primero quitando el filtro que impide que la persona que llama escuche las señales de marcado y luego haciendo algunos ajustes para que el teléfono marcara significativamente más rápido. Básicamente, estaba reprogramando el teléfono, depurando unilateralmente el equipo Western Electric.

El padre de Stew murió cuando él tenía catorce años, y su madre los trasladó a Poughkeepsie, Nueva York. Llegó a un acuerdo con sus profesores de secundaria por el cual arreglaría sus radios y televisores a cambio de no tener que ir a clase. En cambio, pasó un tiempo en una pequeña estación de radio que estaba empezando a funcionar cerca de allí. Nelson “prácticamente la montó todo”, explicó más tarde, conectando los elementos, sintonizando el transmisor, encontrando fuentes de ruido y zumbidos en el sistema. Cuando la estación de radio estaba funcionando, él era el ingeniero principal y, a veces, incluso era el disc jockey. Cada falla en el sistema era una nueva aventura, una nueva invitación a explorar, a probar algo nuevo, a ver qué podía pasar. Para Stewart Nelson, querer averiguar qué podía pasar era la justificación definitiva, más fuerte que la autodefensa o la locura temporal.

Con esa actitud, encajó cómodamente en el Tech Model Railroad Club y el PDP-1. Ya había habido un ávido interés en el “hackeo telefónico” en el club; con Nelson cerca, ese interés realmente podría

florecer. Además de ser un genio técnico, Nelson atacaba los problemas con perseverancia de perro guardián. “Abordaba los problemas tomando acción”, recordó más tarde Donald Eastlake, un hacker de la clase de Nelson. “Era muy persistente. Si lo intentas unas cuantas veces y te rindes, nunca lo conseguirás. Pero si sigues intentándolo... Hay muchos problemas en el mundo que realmente se pueden resolver aplicando el doble o el triple de persistencia que otras personas.

Nelson estaba mostrando una extensión de la ética hacker: si todos actuáramos según nuestro impulso de descubrir, descubriríamos más, produciríamos más, tendríamos el control de más. Naturalmente, el sistema telefónico fue su objeto inicial de exploración en el MIT. Primero el PDP-1 y más tarde el PDP-6 fueron herramientas ideales para usar en estas excursiones. Pero incluso cuando Nelson se embarcó en estos viajes electrónicos, se adhirió a la moralidad hacker no oficial. Se podía llamar a cualquier parte, probar cualquier cosa, experimentar sin fin, pero no se debía hacer para obtener ganancias económicas. Nelson desaprobaba a aquellos estudiantes del MIT que construían "cajas azules" (dispositivos de hardware para hacer llamadas ilegales) con el propósito de estafar a la compañía telefónica. Nelson y los hackers creían que estaban ayudando a la compañía telefónica. Conseguirían líneas telefónicas prioritarias de compañías telefónicas en varios lugares del país y las probarían. Si no funcionaban, lo comunicarían al servicio de reparación correspondiente.

Para hacer esto, por supuesto, había que hacerse pasar por empleados técnicos del sistema telefónico Bell, pero los hackers se volvieron bastante hábiles en eso, especialmente después de leer libros de contrabando como el clásico Principles of Electricity and Electronics Applied to Telephone and Telegraph Work, o Notes on Distant Dialing, o números recientes del Bell System Technical Journal.

Armados con esta información, podrían viajar por todo el mundo y decirle a una operadora: "Llamo desde el panel de pruebas en Hackensack y me gustaría que me pasara a Roma. Estamos tratando de probar el circuito". Ella "escribiría el número", lo que lo llevaría a otro número, y pronto estaría preguntando a una operadora telefónica en Italia qué tiempo hacía allí. O bien, se podía utilizar el PDP-1 en modo Blue Box, lo que le permitía enrutar y redirigir las llamadas hasta que se conectaba con un número de teléfono determinado en Inglaterra, donde las personas que llamaban oirían un cuento infantil antes de dormir, un número al que no se podía acceder desde este país excepto a través de la caja azul.

A mediados de los años sesenta, la compañía telefónica estaba estableciendo su sistema de números gratuitos con código de área 800. Naturalmente, los hackers lo sabían. Con precisión científica, intentarían cartografiar estos reinos indocumentados: excursiones a territorio-800 podía enviarte a lugares extraños, desde las Islas Vírgenes hasta Nueva York. Finalmente, alguien de la compañía telefónica llamó a la línea cercana a la computadora y preguntó qué eran esas cuatrocientas llamadas a lugares que, en lo que a la compañía telefónica concernía, no existían. La desafortunada sucursal de Cambridge de la compañía telefónica ya había tenido que lidiar con el MIT antes, y lo haría de nuevo: en un momento dado, irrumpieron en el noveno piso de Tech Square y exigieron que los hackers les mostraran la caja azul. Cuando los hackers señalaron el PDP-6, los frustrados funcionarios amenazaron con llevarse la máquina entera, hasta que los hackers desconectaron la interfaz telefónica y se la entregaron.

Aunque el interés inicial de Nelson en el PDP-1 era su potencial para hackear teléfonos, se volvió más versátil con él y, con el tiempo, estaba programando todo tipo de cosas. Cuanto más

programaba, mejor se volvía, y cuanto mejor se volvía, más quería programar. Se sentaba junto a la consola de la máquina mientras un estudiante de posgrado se las arreglaba con un programa, y él le daba unos golpecitos en la espalda, lo que sólo conseguía que el estudiante de posgrado se las arreglara aún más, y finalmente estallaba: “Si resuelvo ese problema por ti, ¿me dejarías la computadora?”. El estudiante de posgrado, que probablemente había estado tratando de resolver el problema durante semanas, estaría de acuerdo, sin creer realmente que ese tipo peculiar pudiera resolverlo, pero Nelson ya lo estaría alejando, sentándose en la consola, abriendo el programa de edición “TECO” y tecleando código a un ritmo vertiginoso. En cinco minutos, habría terminado, saltando para imprimirlo en el teletipo Modelo 33 cerca de la máquina, y en un movimiento apresurado arrancaría el papel de la impresora de línea, correría de vuelta a la máquina, sacaría la cinta con el programa del estudiante de posgrado y lo enviaría. Luego haría su propio hackeo.

No conocía límites. Utilizaba tanto el PDP-1 en la sala Kluge como la máquina más nueva en el Proyecto MAC. Cuando otros utilizaban el PDP-1 y su limitado conjunto de instrucciones, tal vez se quejaban de tener que utilizar varias instrucciones para una operación sencilla y luego averiguar las subrutinas para ejecutar los programas. Nelson podía escribir código como los mejores, pero quería más instrucciones en la máquina. Poner una instrucción en el propio ordenador (en el hardware) es una operación bastante complicada. Cuando el TX-0 recibía sus nuevas instrucciones, tenía que apagarse durante un tiempo hasta que sacerdotes oficiales, entrenados casi al nivel del Papa, realizaran la cirugía cerebral necesaria. Esto parecía lógico: ¿quién esperaría que una universidad permitiera a estudiantes de primer año manipular las delicadas partes de un ordenador increíblemente caro?

Nadie. De hecho, Dan Edwards, uno de los estudiantes de posgrado de Minsky que había trabajado en el hacking de Spacewar, se había erigido en protector del hardware. Según Gosper, Edwards había declarado que «¡cualquiera que haga algo tan sencillo como cambiar una cinta en la máquina de escribir va a ser expulsado permanentemente de este lugar!». Pero a los hackers no les importaba lo que la universidad permitía o no permitía. Lo que Dan Edwards pensaba que era aún menos preocupante: su posición de autoridad, como la de la mayoría de los burócratas, se consideraba un accidente.

Nelson pensó que añadir una instrucción de «añadir a la memoria» mejoraría la máquina. Tal vez llevaría meses hacerlo por los canales, y si lo hacía él mismo aprendería algo sobre la forma en que funcionaba el mundo. Así que una noche Stewart Nelson convocó espontáneamente la Sociedad de Cableado de Computadoras de Medianoche. Se trataba de una organización totalmente ad hoc que, cuando el curso de la historia lo requiriera, eludiría las regulaciones del Instituto Tecnológico de Massachusetts contra la manipulación no autorizada de computadoras costosas. La MCWS, que esa noche estaba formada por Nelson, un estudiante trabajador y varios transeúntes interesados, abrió el gabinete y procedió a recablear el PDP-1. Nelson fusionó un par de diodos entre las salidas de la línea “add” y la línea “store” del decodificador de instrucciones, y obtuvo un nuevo código de operación, que presumiblemente respaldaba todas las instrucciones anteriores. Luego procedió a volver a ensamblar la máquina hasta dejarla en un estado aparentemente prístino.

Los hackers probaron la máquina esa noche y funcionó bien. Pero al día siguiente, una usuaria oficialmente autorizada llamada Margaret Hamilton se presentó en el noveno piso para trabajar en algo llamado modelo Vortex para un proyecto de simulación meteorológica en el que estaba trabajando. Margaret Hamilton estaba comenzando una carrera de programación, que la llevaría a encargarse de las computadoras de a bordo en el viaje lunar Apollo, y el programa Vortex en ese

momento era un programa muy importante para ella. Ella era muy consciente de la actitud juguetona de los hackers en el noveno piso, y era moderadamente amigable con algunos de ellos, aunque eventualmente se mezclarían en una personalidad colectiva en su memoria: un joven desaliñado, aunque educado, cuyo amor por la computadora lo había hecho perder toda razón.

El ensamblador que Margaret Hamilton utilizó con su programa Vortex no era el ensamblador MIDAS escrito por los hackers, sino el sistema DECAL proporcionado por DEC que los hackers consideraban absolutamente horrible. Así que, por supuesto, Nelson y el MCWS, al probar la máquina la noche anterior, no habían utilizado el ensamblador DECAL. Nunca habían considerado siquiera la posibilidad de que el ensamblador DECAL accediera al código de instrucciones de una manera diferente a la de MIDAS, una manera que se vio afectada en mayor grado por la ligera caída de voltaje directo creada por la adición de dos diodos entre la línea de adición y la línea de almacenamiento. Margaret Hamilton, por supuesto, no sabía que el PDP-1 había sido sometido a una cirugía la noche anterior. Así que no supo inmediatamente la razón por la que su programa Vortex, después de que lo introdujera con el ensamblador DECAL... se rompió. Dejó de funcionar. Murió. Misteriosamente, un programa perfectamente bueno había fracasado. Aunque los programas a menudo hacían eso por varias razones, esta vez Margaret Hamilton se quejó de ello, y alguien investigó por qué, y alguien más denunció a la Midnight Computer Wiring Society. Así que hubo repercusiones. Reprimendas.

Ése no fue el fin de la Midnight Computer Wiring Society. Edwards y los de su calaña no podían quedarse despiertos toda la noche para vigilar las máquinas. Además, Minsky y los demás a cargo del Proyecto MAC sabían que las actividades nocturnas de los hackers se estaban convirtiendo en un curso práctico de posgrado sobre diseño lógico y habilidades de hardware. En parte porque Nelson y los demás se volvieron lo suficientemente buenos como para que desastres como el Gran Golpe al Programa Margaret Hamilton fueran menos probables de ocurrir, la prohibición oficial del laboratorio de IA contra la manipulación de hardware fue desapareciendo gradualmente hasta convertirse en una de esas leyes anticuadas que nadie se molesta en eliminar de los libros, como un estatuto que te prohíbe golpear a un caballo en público los domingos. Finalmente, la Midnight Computer Wiring Society se sintió lo suficientemente libre para cambiar instrucciones, hacer nuevas conexiones de hardware e incluso conectar el ordenador a las luces de la habitación del noveno piso, de modo que cuando encendías el programa de edición de texto TECO, las luces se atenuaban automáticamente para que pudieras leer la pantalla CRT con mayor facilidad.

Este último hackeo tuvo una consecuencia inesperada. El editor de TECO hacía sonar una campana en el teletipo para avisar cuando el usuario cometía un error. Normalmente esto no era un problema, pero algunos días la máquina se volvía inestable y era extremadamente sensible a las variaciones de la línea eléctrica, como las generadas por la campana del teletipo. En esas ocasiones, cuando alguien cometía un error con TECO, sonaba la campana y la máquina se volvía aleatoria. El ordenador se descontrolaba; escribía espasmódicamente, haciendo sonar la campana y, lo más inquietante, encendiendo y apagando las luces de la habitación. ¡El ordenador se había vuelto loco! ¡Un Armagedón de ciencia ficción!

Los hackers lo consideraron extremadamente gracioso.

Las personas a cargo del laboratorio, en particular Marvin Minsky, eran muy comprensivos con estas cosas. Marvin, como lo llamaban los hackers (invariablemente se llamaban entre ellos por el apellido), sabía que la Ética Hacker era lo que mantenía al laboratorio productivo y no iba a alterar

uno de los componentes cruciales del hackerismo. Por otro lado, estaba Stew Nelson, que siempre se saltaba las normas, un tipo muy sensible que se puso aún más nervioso cuando finalmente lo pillaron con las manos en la masa hackeando un teléfono. Había que hacer algo. Así que Minsky llamó a su buen amigo Ed Fredkin y le dijo que tenía un problema con un joven de diecinueve años increíblemente brillante que tenía tendencia a meterse en líos sofisticados. ¿Podría Fredkin contratarlo?

• • • • • • •

Además de ser un amigo íntimo de Marvin Minsky y el fundador de Information International Incorporated (Triple-I), Ed Fredkin se consideraba el mejor programador del mundo.

Fredkin, un hombre de pelo oscuro y cálidos ojos castaños detrás de unas gafas que descansaban sobre una nariz con un ligero gancho intelectual, nunca había terminado la universidad. Había aprendido informática en la Fuerza Aérea en 1956, como uno de los primeros hombres que trabajaron en el sistema informático de defensa aérea SAGE, que entonces se consideraba el sistema más complicado conocido por el hombre. Fredkin y otros diecinueve comenzaron un curso intensivo en el campo en ciernes de la computación: tambores de memoria, lógica, comunicaciones y programación. Fredkin recordó más tarde, con su voz tranquilizadora de narrador de historias: "Después de una semana, todos abandonaron el curso, menos yo".

Ed Fredkin no se enamoró perdidamente de las computadoras como Kotok, Samson, Greenblatt o Gosper; en algunos sentidos era un hombre muy mesurado, demasiado políglota intelectual como para obsesionarse únicamente con las computadoras. Pero sentía una intensa curiosidad por ellas, por lo que después de dejar el servicio aceptó un trabajo en el Laboratorio Lincoln, afiliado al MIT, donde pronto se ganó la reputación de ser el mayor fastidioso de la programación. Siempre podía idear algoritmos originales, algunos de los cuales se hicieron conocidos como protocolos de programación estándar. También fue uno de los primeros en ver la importancia del PDP-1: lo conocía antes de que se construyera el prototipo y encargó el primero. Bolt Beranek y Newman lo convenció de no comprarlo, y que en cambio la contrataron para programar la máquina y escribir un ensamblador. Fredkin lo hizo y modestamente lo consideró una obra maestra de la programación. Además del trabajo de sistemas, Fredkin se dedicó al tipo de hacking matemático que más tarde sería el fuerte de Bill Gosper, y realizó algunas de las primeras teorías sobre autómatas. Pero al no ser un hacker puro (tenía instintos comerciales y una familia que mantener), dejó BBN para fundar su propia empresa, Information International, que realizaría todo tipo de resolución de problemas digitales y consultas informáticas especiales. La empresa finalmente tuvo su sede en Los Ángeles, pero durante mucho tiempo tuvo instalaciones en Tech Square, dos pisos debajo del PDP-6.

Fredkin estaba encantado con la comunidad hacker de Tech Square; habían llevado el hacking más allá de su estado anterior, que se encontraba solo a tiempo parcial en los pocos lugares del mundo (como el MIT, el DEC, el Ejército, BBN) donde las computadoras eran accesibles para personas para quienes la informática era un fin en sí misma. Alrededor del MIT, el hacking era a tiempo completo. Fredkin llegó a querer a los hackers: podía hablar su idioma y admirar su trabajo. A veces los acompañaba en sus excursiones a Chinatown y en esas ocasiones las discusiones podían volverse bastante espontáneas. Muchos de los hackers eran ávidos fanáticos de la ciencia ficción (nótese el origen de Spacewar), pero Fredkin fue capaz de vincular las maravillas de Heinlein y Asimov con el trabajo que estaban haciendo los hackers: convertir las computadoras en sistemas poderosos y construir una base de software para la inteligencia artificial. Fredkin tenía un talento

para despertar su imaginación, como lo hizo cuando reflexionó sobre que un día la gente tendría pequeños robots en sus cabezas que cortarían el cabello cuando alcanzara la longitud precisa para el peinado deseado. (Fredkin causaría un alboroto nacional cuando repitió esta predicción en un programa de entrevistas de televisión). Sin embargo, por mucho que Fredkin admirara a los hackers, seguía pensando que era el mejor programador. Aunque la ética hacker fomentaba el esfuerzo colectivo para la mejora general, cada hacker quería ser reconocido como un mago, y los programas rápidos y los esfuerzos de creación de código fulgurante se exhibían y discutían con entusiasmo. Era un gran estímulo para el ego estar en la cima de la colina de los hackers, donde Fredkin se consideraba a sí mismo. Hackear, para Fredkin, era sobre todo un orgullo por la artesanía.

“Nunca me había topado con nadie que pudiera superarme en código, en ningún sentido”, recordó Fredkin más tarde. “Pero estaba muy claro que Nelson podía”. Nelson era un genio en su conocimiento de la informática, innovador en su enfoque, increíblemente intenso en el abordaje de los problemas y capaz de una concentración sobrehumana. Fredkin contrató al joven hacker por recomendación de Minsky, y no tardó mucho en darse cuenta de que incluso en un lugar donde la programación excepcional era algo común, Nelson era algo especial, una ola humana de programadores de un solo hombre. Por supuesto, como Triple-I estaba en Tech Square, Nelson también podía pasar el rato en el laboratorio de IA en el noveno piso y hacer el trabajo de varios programadores allí también. Pero eso no era motivo de queja; cuando Fredkin lo necesitaba, Nelson casi siempre podía inventar algo mágico.

Había un proyecto de programación en particular, una tarea en el DEC PDP-7, en la que Fredkin quería que Nelson trabajara, pero por alguna razón Nelson no lograba motivarse. La compañía de Fredkin también necesitaba al mismo tiempo un diseño para una interfaz entre un determinado ordenador y una unidad de disco para el almacenamiento de datos. Fredkin consideraba que este último era un proyecto que llevaría seis meses y quería que la otra tarea se hiciera primero. Nelson le prometió que obtendría algunos resultados durante el fin de semana. El lunes siguiente, Nelson llegó con un trozo de papel gigante casi completamente cubierto de pequeños garabatos, largas líneas que conectaban un bloque de garabatos con otro y evidencia de borrado y sobreescritura frenéticos. No era el programa PDP-7 que Fredkin había pedido, sino la interfaz completa de la unidad de disco. Nelson lo había probado como una escapatoria constructiva de la tarea asignada. La compañía de Fredkin construyó el equipo directamente a partir de ese trozo de papel, y funcionó.

Fredkin estaba encantado, pero todavía quería que el problema del PDP-7 también se resolviera. Así que dijo: “Nelson, tú y yo nos sentaremos y programaremos esto juntos. Tú escribes esta rutina y yo escribiré aquella”. Como no tenían un PDP-7 cerca, se sentaron en las mesas para escribir su código ensamblador predepurado. Comenzaron a trabajar. Tal vez fue en ese momento cuando Ed Fredkin se dio cuenta, de una vez por todas, de que no era el mejor programador del mundo. Nelson corría como si fuera solo una cuestión de cuán rápido pudiera plasmar sus garabatos en el papel. Fredkin finalmente se sintió invadido por la curiosidad y miró el programa de Nelson. No lo podía creer. Era extraño. Totalmente no obvio, un mosaico de subrutinas entrelazadas. Y estaba claro que funcionaría. “Stew”, estalló Fredkin, “¿por qué diablos lo escribes de esta manera?” Nelson explicó que una vez había escrito algo similar en el PDP-6 y, en lugar de pensar en ello, simplemente estaba transliterando las rutinas anteriores, de memoria, al código PDP-7. Un ejemplo perfecto de cómo funcionaba la mente de Nelson. Tenía su propio comportamiento hasta el punto de poder grabar instrucciones mentales y minimizar el trabajo que hacía.

Era claramente un enfoque que se adaptaba mejor al trabajo con máquinas que a la interacción humana. Nelson era extremadamente tímido y Fredkin probablemente actuaba como una figura paterna para el joven hacker. Más tarde recordaría haberse sorprendido un día cuando Nelson entró en su oficina y dijo: "¿Adivina qué? ¡Me voy a casar!".

Fredkin habría juzgado que Nelson no sabía cómo pedirle una cita a una mujer, y mucho menos hacerle una propuesta de matrimonio. "¡Fantástico!", dijo. "¿Quién es la chica afortunada?".

"Oh, no sé", dijo Nelson. "Simplemente decidí que sería una buena cosa para hacer".

Quince años después, Nelson todavía estaba en modo soltero.

Si bien las mujeres pueden no haber sido una gran presencia en su vida, Nelson sí tenía la compañía de otros hackers. Se mudó a una casa con Gosper y otras dos personas. Aunque esta “casa de hackers” estaba en el cercano Belmont, luego se mudó a Brighton, Nelson se resistió a comprar un auto. No soportaba conducir. “Requiere demasiado procesamiento lidiar con la carretera”, explicaría más tarde. Tomaba el transporte público, o pedía que otro hacker lo llevara, o incluso tomaba un taxi. Una vez que llegaba a Tech Square, estaba bien para las horas: Nelson estaba entre aquellos hackers que habían establecido la rutina de veintiocho horas diarias, seis días a la semana. No se preocupaba por las clases: pensaba que podría conseguir el trabajo que quisiera, tuviera un título o no, por lo que nunca se matriculó de nuevo.

Nelson era completamente una criatura de la ética hacker, y la influencia de su comportamiento fue un factor que contribuyó al crecimiento cultural y científico del laboratorio de IA. Si Minsky necesitaba que alguien le señalara por qué cierta subrutina no funcionaba, acudiría a Nelson. Mientras tanto, Nelson estaría por todas partes. Trabajaba para Fredkin, hacía trabajos de sistemas con Greenblatt, hackeaba pantallas con Gosper y creaba todo tipo de cosas extrañas. Hackeó una conexión extraña entre la computadora Triple-I en el séptimo piso y la PDP-6 en el noveno, que enviaba señales entre un osciloscopio en una línea y una cámara de televisión en otra. Hizo todo tipo de nuevos hackeos telefónicos. Y, nuevamente más con el ejemplo que organizando, fue un líder en el sagrado arte negro de hackear cerraduras.

• • • • • • •

“Hackear cerraduras” era la hábil solución de cerraduras físicas, ya sea en puertas, archivadores o cajas fuertes. Hasta cierto punto, la práctica era una tradición del MIT, especialmente en torno a TMRC. Pero una vez que se combinó con la ética hacker, el hacking de cerraduras se convirtió más en una cruzada que en un juego ocioso, aunque el desafío lúdico de superar obstáculos artificiales contribuyó a la popularidad del hacking de cerraduras.

Para un hacker, una puerta cerrada es un insulto, y una puerta cerrada es un ultraje. Así como la información debe transportarse de forma clara y elegante dentro de una computadora, y así como el software debe difundirse libremente, los hackers creían que la gente debería tener acceso a archivos o herramientas que pudieran promover la búsqueda de hackers para descubrir y mejorar la forma en que funciona el mundo. Cuando un hacker necesitaba algo que lo ayudara a crear, explorar o arreglar, no se molestaba con conceptos tan ridículos como los derechos de propiedad.

Digamos que estás trabajando en el PDP-6 una noche, y se estropea. Revisas sus entrañas y descubres que necesita una pieza. O tal vez necesites una herramienta para instalar una pieza. Luego descubres que lo que necesitas (un disco, una cinta, un destornillador, un soldador, un CI (circuito integrado) de repuesto) está encerrado en algún lugar. Un millón de dólares en hardware

desperdiciado e inactivo, porque el mago del hardware que sabe cómo arreglarlo no puede acceder al circuito integrado de setenta y cinco centavos, o al osciloscopio guardado en una caja fuerte. De modo que los hackers se las arreglaban para obtener las llaves de estos casilleros y estas cajas fuertes. De modo que podían hacerse con las piezas, mantener los ordenadores en funcionamiento, reemplazar cuidadosamente lo que habían robado y volver al trabajo.

Como dijo más tarde un hacker llamado David Silver, era "una guerra ultrainteligente... había administradores que tenían cerraduras de alta seguridad y bóvedas donde almacenar las llaves, y tarjetas de registro para crear llaves. Y se sentían seguros, como si estuvieran acerrojando todo y controlando las cosas e impidiendo que la información fluyera por el camino equivocado y que las cosas fueran robadas. Luego estaba el otro lado del mundo donde la gente sentía que todo debería estar disponible para todos, y estos hackers tenían kilos y kilos y kilos de llaves que les permitían entrar en todos los lugares imaginables. Las personas que hacían esto eran muy éticas y honestas y no usaban este poder para robar o herir. Era una especie de juego, en parte por necesidad y en parte por ego y diversión... Si estabas en el círculo interior correcto, en el punto más alto, podías obtener la combinación de cualquier caja fuerte y tener acceso a cualquier cosa".

La adquisición básica de todo hacker para abrir cerraduras era una llave maestra. La llave maestra adecuada abriría las puertas de un edificio o de un piso de un edificio. Incluso mejor que una llave maestra era una llave maestra mayor, una especie de llave maestra de llaves maestras; una de esas pequeñas cosas podía abrir quizás dos tercios de las puertas del campus. Al igual que el phreaking, el hacking de cerrojos requería persistencia y paciencia. Así que los hackers hacían excursiones nocturnas, desatornillando y quitando cerraduras de puertas. Luego desmontaban cuidadosamente las cerraduras. La mayoría de las cerraduras se podían abrir con varias combinaciones de llave diferentes; así que los hackers desarmaban varias cerraduras en el mismo pasillo para determinar qué combinación aceptaban en común. Luego intentaban hacer una llave con la forma de esa combinación particular.

Podría ser que la llave maestra tuviera que hacerse a partir de "piezas en blanco" especiales, no disponibles para el público en general. (Este suele ser el caso de las llaves maestras de alta seguridad, como las que se usan en el trabajo de defensa). Esto no detuvo a los hackers, porque varios de ellos habían tomado cursos por correspondencia para calificar para la certificación de cerrajeros; Se les permitió oficialmente comprar esas llaves en blanco restringidas. Algunas llaves eran de tan alta seguridad que ni siquiera los cerrajeros con licencia podían comprar llaves en blanco para ellas; para duplicarlas, los hackers hacían llamadas a medianoche al taller de máquinas, un espacio de trabajo en una esquina del noveno piso donde un hábil artesano del metal llamado Bill Bennett trabajaba durante el día en materiales como brazos de robot. Trabajando desde cero, varios hackers hicieron sus propias llaves en blanco en el taller de máquinas.

La llave maestra era más que un medio para un fin; era un símbolo del amor de los hackers por el libre acceso. En un momento dado, los hackers del TMRC incluso consideraron enviar una llave maestra del MIT a cada estudiante de primer año entrante como incentivo para el reclutamiento. La llave maestra era una espada mágica para alejar el mal. El mal, por supuesto, era una puerta cerrada. Incluso si no había herramientas detrás de puertas cerradas, las cerraduras simbolizaban el poder de la burocracia, un poder que eventualmente se usaría para evitar la implementación completa de la Ética Hacker. Las burocracias siempre se vieron amenazadas por personas que querían saber cómo funcionaban las cosas. Los burócratas sabían que su supervivencia dependía de mantener a la gente en la ignorancia, utilizando medios artificiales (como cerraduras) para mantener a la gente bajo

control. Así que cuando un administrador subía la apuesta en esta guerra instalando una nueva cerradura o comprando una caja fuerte de Clase Dos (certificada por el gobierno para material clasificado), los hackers trabajaban inmediatamente para romper la cerradura y abrir la caja fuerte. En este último caso, iban a un depósito de excedentes superultratecnológicos en Taunton, encontraban una caja fuerte de Clase Dos similar, la llevaban al noveno piso y la abrían con sopletes de acetileno para averiguar cómo funcionaban las cerraduras y los pestillos.

Con todo este hackeo de cerrojos, el laboratorio de IA era la pesadilla de un administrador. Russ Noftsker lo sabía; él era el administrador. Había llegado a Tech Square en 1965 con un título de ingeniería de la Universidad de México, un interés en la inteligencia artificial y un amigo que trabajaba en el Proyecto MAC. Conoció a Minsky, cuyo principal estudiante de posgrado y administrador, Dan Edwards, acababa de salir del laboratorio. Minsky, conocido por su desinterés en la administración, necesitaba a alguien que se encargara del papeleo del laboratorio de IA, que finalmente se separaría del Proyecto MAC para convertirse en una entidad independiente con su propia financiación gubernamental. Así que Marvin contrató a Noftsker, quien a su vez contrató oficialmente a Greenblatt, Nelson y Gosper como hackers a tiempo completo. De algún modo, Noftsker tenía que mantener este circo electrónico en línea con los valores y la política del Instituto.

Noftsker, un rubio de complexión compacta, rasgos fruncidos y ojos azules que podían parecer soñadores o preocupados, no era ajeno a las raras hazañas tecnológicas: cuando estaba en la escuela, había hackeado explosivos con un amigo. Trabajaban para una empresa de alta tecnología y cobraban sus salarios en primacord (un material altamente combustible) o dinamita, y hacían explosiones en cuevas para ver cuántas arañas podían hacer estallar, o para ver cuánto primacord hacía falta para partir por la mitad un bidón de sesenta y cinco galones. Un amigo de Noftsker estaba derritiendo una noche treinta libras de TNT en el horno de su madre cuando se incendió. El horno y el refrigerador se derritieron y el chico se vio en la incómoda situación de tener que ir a la casa del vecino de al lado y decirle: "Disculpe, creo que sería una buena idea que se mudara un poco más abajo...". Noftsker sabía que había tenido suerte de sobrevivir aquellos días; sin embargo, según Gosper, más tarde Noftsker urdió un plan para limpiar la nieve de su acera con primacord, hasta que su esposa le puso fin a la idea. Noftsker también compartía la aversión de los hackers al humo del cigarrillo y a veces expresaba su desagrado disparando un chorro de oxígeno puro de un bote que tenía para ese propósito; el fumador asombrado veía cómo su cigarrillo estallaba en una feroz mancha naranja. Obviamente, Noftsker entendía el concepto del extremismo tecnológico para mantener un ambiente agradable.

Por otro lado, Noftsker estaba al mando, maldita sea, y parte de su trabajo consistía en mantener a la gente fuera de las zonas cerradas y mantener privada la información confidencial. Se jactaba, amenazaba, mejoraba las cerraduras y encargaba cajas fuertes, pero sabía que, en última instancia, no podría imponerse por la fuerza. Por ingenuo que fuera el pensamiento en el mundo real, los hackers creían que los derechos de propiedad no existían. En lo que respecta al noveno piso, ese era efectivamente el caso. los hackers podían entrar en cualquier lugar, como Noftsker vio gráficamente un día cuando llegó una nueva caja fuerte con una cerradura a prueba de ganzúas las veinticuatro horas y alguien cerró la caja fuerte sin darse cuenta y giró el dial antes de que Noftsker consiguiera la combinación del fabricante. Uno de los hackers, que era cerrajero registrado, se ofreció a ayudar y abrió la caja fuerte en veinte minutos.

Entonces, ¿qué debía hacer Noftsker?

“Erigir barreras [aumentaría] el nivel del desafío”, explicaría Noftsker más tarde. “Así que el truco era llegar a una especie de acuerdo tácito –que ‘Esta línea, por imaginaria que sea, está fuera de los límites’– para dar a la gente que sentía que necesitaba algo de privacidad y seguridad la sensación de que realmente la tenían. Y si alguien violaba esos límites, la violación sería tolerada siempre y cuando nadie lo supiera. Por lo tanto, si ganabas algo arrastrándote por el muro para entrar en mi oficina, nunca tenías que decir nada al respecto”.

Desarme unilateral. Dar a los hackers vía libre para ir a donde quisieran en sus exploraciones, tomar lo que quisieran para ayudarlos en sus divagaciones electrónicas y sesiones improvisadas de informática... siempre y cuando no anduvieran por ahí alardeando de que el emperador burocrático estaba desnudo. De esa manera, Noftsker y la administración a la que representaba podían mantener cierta dignidad mientras los hackers podían fingir que la administración no existía. Iban a donde querían, entraban en las oficinas a través del espacio de acceso creado por el techo artificial, quitaban una placa del techo y se dejaban caer en sus destinos: comandos con amigos-lápiz en los bolsillos de sus camisas. Un hacker se lastimó la espalda una noche cuando el techo se derrumbó y cayó en la oficina de Minsky. Pero más a menudo, la única evidencia que Noftsker encontraba era la ocasional huella en su pared. Y, por supuesto, a veces entraba en su oficina cerrada y descubría a un hacker dormitando en el sofá.

Sin embargo, algunas personas nunca pudieron tolerar la Ética Hacker. Aparentemente, una de ellas era el artesano del taller de máquinas Bill Bennett. Aunque era miembro del TMRC, no era en absoluto un hacker: su lealtad no era hacia la facción de Señales y Energía, sino hacia lo que Gosper llamaba la “Subcultura de Construyamos Pequeños Dispositivos Físicos Precisos en Miniatura”. Era un buen chico de Marietta, Georgia, y tenía un respeto casi religioso por sus herramientas. En su tierra natal, la tradición consideraba que las herramientas eran objetos sagrados, cosas que se cuidan y se conservan y, en última instancia, se entregan a los nietos. “Soy un fanático”, explicaría más tarde. “Una herramienta debe estar en su lugar correcto, limpia y lista para usar”. Así que no sólo encerraba todas sus herramientas bajo llave, sino que prohibía a los hackers incluso entrar en su espacio de trabajo, que acordonaba colocando una valla de cuerda y pintando rayas en el suelo.

Bennett no podía evitar el resultado inevitable de trazar una línea y decirles a los hackers que no podían cruzar. Cuando entraba, veía que se habían utilizado sus herramientas y se quejaba con Minsky. Amenazaba con marcharse; Noftsker recuerda que amenazaba con poner trampas explosivas en su zona. Exigía especialmente que Minsky se vengara de Nelson, a quien, al parecer, consideraba el peor infractor. Minsky o Noftsker podían hacer el gesto de reprender a Nelson, pero en privado consideraban que el drama era bastante divertido. Con el tiempo, a Noftsker se le ocurrió la idea de dar a cada hacker su propia caja de herramientas, y que cada uno fuera responsable de sus propias herramientas, pero eso no funcionó demasiado bien. Cuando un hacker quiere ajustar algo en una máquina, o quiere crear un hack rápido de hardware, usa cualquier cosa que tenga a mano, ya sea de un amigo o una de las posesiones más mimadas de Bill Bennett. Una vez, Nelson usó esto último, un destornillador, y en el transcurso de su trabajo lo marcó un poco. Cuando Bennett llegó al día siguiente y encontró un destornillador dañado, fue directo a por Nelson.

Nelson era normalmente muy callado, pero a veces explotaba. Gosper lo describió más tarde: “Nelson era un discutidor increíble. Si acorralabas a Nelson, pasaba de ser un tipo tímido a un completo salvaje”. Así que, como Gosper recordó más tarde, Nelson y Bennett se enzarzaron en una discusión a gritos, y durante la misma Nelson dijo que el destornillador estaba casi “agotado”, de todos modos.

¿Agotado? Fue una filosofía increíblemente ofensiva para Bennett. “Esto hizo que saliera humo de las orejas de Bennett”, contó Gosper más tarde. “Simplemente explotó”. Para personas como Bennett, las cosas no se transmiten de persona a persona hasta que ya no son útiles. No es como un programa de ordenador que escribes y pules, y luego lo dejas para que otros, sin pedirte permiso, trabajen en él, añadan nuevas funciones, lo reformulen a su propia imagen y luego lo dejas para que la siguiente persona lo mejore, repitiéndose el ciclo una y otra vez cuando alguien crea desde cero un nuevo y maravilloso programa para hacer lo mismo. Puede que eso fuera lo que creían los hackers, pero Bill Bennett pensaba que las herramientas eran algo que uno poseía, algo privado. Estos hackers en realidad pensaban que una persona tenía derecho a utilizar una herramienta sólo porque pensaba que podía hacer algo útil con ella. Y cuando terminaban, la tiraban a la basura, diciendo que estaba... ¡agotada!

Teniendo en cuenta estas filosofías diametralmente opuestas, no fue una sorpresa que Bennett estallara en cólera con Nelson. Bennett diría más tarde que sus arrebatos siempre eran rápidos y seguidos de la buena voluntad habitual que existía entre él y los hackers. Pero Nelson diría más tarde que en ese momento había tenido miedo de que el maquinista pudiera hacerle daño físico. Unas noches después, Nelson quiso realizar algunos ajustes completamente no autorizados en la fuente de alimentación de una computadora en el séptimo piso de Tech Square y necesitaba un destornillador grande para hacerlo. Naturalmente, fue al armario cerrado con llave de Bennett para buscar la herramienta. De alguna manera, los disyuntores de la fuente de alimentación estaban en un estado precario y Nelson recibió una gran descarga eléctrica. Nelson sobrevivió sin problemas, pero la descarga derritió la punta del destornillador.

Al día siguiente, Bill Bennett regresó a su oficina y encontró su destornillador destrozado con un cartel que decía USADO.

Capítulo 6. Ganadores y perdedores

En 1966, cuando David Silver tomó su primer viaje en ascensor hasta el noveno piso de Tech Square, el laboratorio de IA era una comunidad de muestra, que trabajaba bajo los sagrados preceptos de la Ética Hacker. Después de una gran cena china, los hackers se quedaban hasta el amanecer, congregándose alrededor del PDP-6 para hacer lo que era más importante en el mundo para ellos. Iban y venían con sus impresiones y sus manuales, charlando con quienquiera que estuviera usando la terminal en ese momento, apreciando el estilo con el que el programador escribía su código. Obviamente, la clave del laboratorio era la cooperación y una creencia conjunta en la misión de hackear. Estas personas estaban apasionadamente involucradas en la tecnología, y tan pronto como los vio, David Silver quiso pasar todo su tiempo allí.

David Silver tenía catorce años. Estaba en sexto grado, había sido expulsado dos veces. Apenas sabía leer. Sus compañeros de clase a menudo se burlaban de él. Más tarde, la gente reflexionaría sobre su problema: la dislexia. Silver simplemente diría que "no le interesaban" los profesores, los estudiantes ni nada de lo que sucedía en la escuela. Le interesaba construir sistemas.

Desde que tenía seis años, iba regularmente al depósito de chatarra de Eli Heffron en Cambridge (donde los hackers de TMRC también hurgaban) y recuperaba todo tipo de cosas fascinantes. Una vez, cuando tenía unos diez años, regresó con una antena parabólica, la desarmó y la reconstruyó para que pudiera captar sonidos: la manipuló como un reflector parabólico, le colocó un micrófono y pudo captar conversaciones a miles de metros de distancia. La mayoría de las veces escuchaba coches lejanos, pájaros o insectos. También construyó muchos equipos de audio y se dedicó a la fotografía time-lapse. Luego se interesó por las computadoras.

Su padre era científico, amigo de Minsky y profesor en el MIT. En su oficina tenía una terminal conectada al sistema de tiempo compartido compatible del IBM 7094. David empezó a trabajar con él: su primer programa estaba escrito en LISP y traducía frases inglesas al latín. Después empezó a trabajar en un programa que controlaría un pequeño robot (lo llamó un "bicho") que construyó en casa con viejos relés telefónicos que consiguió en la tienda de Eli. Conectó el bicho a la terminal y, trabajando en lenguaje de máquina, escribió un programa que hacía que el bicho de dos ruedas realmente se arrastrara. David decidió que la robótica era la mejor de todas las actividades: ¿qué podría ser más interesante que fabricar máquinas que pudieran moverse por sí solas, ver por sí solas… pensar por sí solas?

Por eso su visita al laboratorio de IA, organizada por Minsky, fue una revelación. No sólo estas personas estaban tan entusiasmadas con las computadoras como David Silver, sino que una de las principales actividades del laboratorio era la robótica. Minsky estaba sumamente interesado en ese campo. La robótica era crucial para el progreso de la inteligencia artificial; Nos permitió ver hasta dónde podía llegar el hombre para conseguir que las máquinas inteligentes hicieran su trabajo. Muchos de los estudiantes de posgrado de Minsky se interesaron por la teoría de la robótica, elaborando tesis sobre la relativa dificultad de conseguir que un robot hiciera esto o aquello. Los hackers también estaban muy involucrados en el campo, no tanto en la teoría como en la construcción y la experimentación. Los hackers amaban a los robots por las mismas razones que a David Silver. Controlar un robot era un paso más allá de la programación informática en el control del sistema que era el mundo real. Como solía decir Gosper, "¿Por qué deberíamos limitar los ordenadores a las mentiras que la gente les dice a través de los teclados?" Los robots podían salir y descubrir por sí mismos cómo era el mundo.

Cuando se programa un robot para que haga algo, Gosper explicaría más tarde, se obtiene “una especie de gratificación, un impacto emocional, que es completamente indescriptible. Y supera con creces el tipo de gratificación que se obtiene de un programa en funcionamiento. Se obtiene una confirmación física de la corrección de la construcción. Tal vez sea algo así como tener un hijo”.

Un gran proyecto que los hackers completaron fue un robot que podía atrapar una pelota. Utilizando un brazo mecánico controlado por el PDP-6, así como una cámara de televisión, Nelson, Greenblatt y Gosper trabajaron durante meses hasta que el brazo finalmente pudo atrapar una pelota de ping-pong que se le lanzara. El brazo pudo determinar la ubicación de la pelota a tiempo para moverse en posición para atraparla. Era algo de lo que los hackers estaban tremendamente orgullosos, y Gosper quería especialmente ir más allá y comenzar a trabajar en un robot más móvil que realmente pudiera jugar al ping-pong.

"¿Ping-pong para Navidad?", preguntó Minsky a Gosper mientras observaban al robot atrapar pelotas.

El ping-pong, como los restaurantes chinos, era un sistema que Gosper respetaba. Había jugado al juego en el sótano de su casa cuando era niño, y su estilo de ping-pong tenía mucho en común con su estilo de hackeo: ambos se basaban en su amor por lo físicamente improbable. Cuando Gosper golpeó una pelota de ping-pong, el resultado fue algo tan disparatado como un truco de pantalla PDP-6: puso tanto inglés en la pelota que se convocaron fuerzas complejas y contraintuitivas, y no había forma de saber dónde la pelota podía ir a parar. A Gosper le encantaba el efecto de giro, la negación de la gravedad que permitía golpear violentamente una pelota de modo que, en lugar de pasar volando por el extremo de una mesa, de repente se curvara hacia abajo y, cuando el oponente intentara golpearla, la pelota giraría tan furiosamente que volaría hacia el techo. O golpeaba una pelota para aumentar tanto el giro que casi se aplanaba, casi explotando en el aire por la fuerza centrífuga. “Había momentos en los que, en los juegos que estaba jugando”, diría Gosper más tarde, “una pelota hacía algo en el aire, algo no físico, que hacía que los espectadores se quedaran boquiabiertos. He visto cosas inexplicables suceder en el aire. Esos eran momentos interesantes”.

Gosper estuvo obsesionado durante un tiempo con la idea de que un robot jugara el juego. los hackers lograron que el robot sostuviera una paleta y diera un buen manotazo a una pelota lanzada en su dirección. Bill Bennett recordaría más tarde una ocasión en la que Minsky entró en la zona del brazo robótico, iluminada por las luces brillantes que requiere la cámara vidicón; el robot, al ver el resplandor que se reflejaba en la calva cabeza de Minsky, confundió al profesor con una gran pelota de ping-pong y casi lo decapitó.

Gosper quería llegar hasta el final; que el robot estuviera preparado para moverse y hacer tiros inteligentes, tal vez con el efecto sobrenatural de una buena volea de Gosper. Pero Minsky, que en realidad había hecho parte del diseño del hardware para la máquina atrapapelotas, no pensó que fuera un problema interesante. No lo consideraba diferente del problema de derribar misiles del cielo con otros misiles, una tarea que el Departamento de Defensa parecía tener bajo control. Minsky disuadió a Gosper de seguir adelante con el proyecto de ping-pong, y Gosper insistiría más tarde en que ese robot podría haber cambiado la historia.

Por supuesto, la idea de que se considerara un proyecto como ese era emocionante para David Silver. Minsky había permitido que Silver pasara el rato en el noveno piso, y pronto Silver abandonó la escuela por completo para poder pasar su tiempo de manera más constructiva en Tech Square. Como a los hackers les importa menos la edad de las personas que la contribución potencial

de alguien al hacking, David Silver, de catorce años, fue aceptado, al principio como una especie de mascota.

Inmediatamente demostró su valor al ofrecerse voluntario para realizar algunas tareas tediosas de hackeo de cerrojos. Era una época en la que la administración había instalado un nuevo y resistente sistema de cerraduras de alta seguridad. A veces, el adolescente de complexión delgada pasaba una noche entera gateando por falsos techos para desmontar las cerraduras de un pasillo, estudiarlas para ver cómo funcionaba el sistema de control y reconstruirlas minuciosamente antes de que los administradores regresaran por la mañana. Silver era muy bueno trabajando con herramientas de maquinista, y mecanizó una determinada pieza en bruto que podía convertirse en una llave para abrir una nueva cerradura particularmente resistente. La cerradura estaba en una puerta que protegía una habitación con una caja fuerte de alta seguridad que contenía... llaves. Una vez que los hackers llegaron a ese punto, el sistema se "desmoronó", en palabras de Silver.

Silver veía a los hackers como sus maestros: podía preguntarles cualquier cosa sobre computadoras o máquinas y ellos le arrojaban enormes cantidades de conocimiento. Se lo transmitían en la colorida jerga hacker, cargada de extrañas variaciones del idioma inglés que recordaban a los ositos de peluche. Palabras como winnitude, Greenblattful, gronk y foo eran elementos básicos del vocabulario hacker, abreviaturas que utilizaban personas relativamente no verbales para comunicar exactamente lo que tenían en mente.

Silver tenía todo tipo de preguntas. Algunas de ellas eran muy básicas: ¿de qué están hechas las distintas piezas de las computadoras? ¿De qué están hechos los sistemas de control? Pero a medida que se adentraba más en la robótica, descubrió que las preguntas que tenía que hacer eran de doble filo. Tenía que considerar las cosas en términos casi cósmicos antes de poder crear la realidad para un robot. ¿Qué es un punto? ¿Qué es la velocidad? ¿Qué es la aceleración? Preguntas sobre física, preguntas sobre números, preguntas sobre información, preguntas sobre la representación de las cosas... Llegó a un punto, como Silver se dio cuenta más tarde, en el que "se planteaba cuestiones filosóficas básicas como qué soy yo, qué es el universo, qué son las computadoras, para qué se pueden usar y cómo se relaciona todo eso. En ese momento, todas esas preguntas eran interesantes porque era la primera vez que había comenzado a reflexionar y a saber lo suficiente sobre las computadoras, y estaba relacionando funciones biológicas, humanas y animales, y comenzando a relacionarlas con la ciencia, la tecnología y las computadoras. Comencé a darme cuenta de que existía esta idea de que se podían hacer cosas con las computadoras que son similares a las cosas que hacen los seres inteligentes".

El gurú de Silver era Bill Gosper. A menudo iban a uno de los dormitorios a jugar al ping-pong, a comer comida china o a hablar sobre computadoras y matemáticas. Mientras tanto, Silver absorbía conocimientos en este Xanadú sobre Cambridge. Era una escuela que nadie más conocía y, por primera vez en su vida, era feliz.

La computadora y la comunidad que la rodeaba lo habían liberado, y pronto David Silver se sintió preparado para trabajar seriamente en el PDP-6. Quería escribir un programa grande y complicado: quería modificar su pequeño robot "bicho" para que usara la cámara de televisión para "buscar" cosas que la gente arrojaría al suelo. Los hackers no se inmutaron por el hecho de que nadie, ni siquiera gente experimentada con acceso a todo tipo de equipos sofisticados, había hecho algo similar. Silver lo hizo con su estilo inquisitivo habitual, yendo a diez o veinte hackers y preguntando a cada uno sobre una sección específica de la parte de visión del programa. Tom Sawyer, un experto

en alta tecnología, pintando una valla con código ensamblador. Los problemas de hardware, se los preguntaba a Nelson. Los problemas de sistemas, a Greenblatt. Las fórmulas matemáticas, a Gosper. Y luego pedía a la gente que le ayudara con una subrutina para ese problema. Cuando consiguió todas las subrutinas, trabajó para armar el programa y obtuvo su programa de visión. El bicho en sí medía un pie de largo y siete pulgadas de ancho, y estaba hecho de dos pequeños motores atados con un arnés de plástico. Tenía ruedas erectoras en cada extremo, una barra erectora que atravesaba la parte superior y barras de soldadura de cobre que sobresalían por delante, como un par de astas. Francamente, parecía un trozo de chatarra. Silver utilizó una técnica llamada "sustracción de imágenes" para que la computadora supiera dónde estaba el insecto en cualquier momento: la cámara siempre escanearía la escena para ver qué se había movido y notaría cualquier cambio en su imagen. Mientras tanto, el insecto se movería aleatoriamente hasta que la cámara lo detectara y la computadora lo dirigiera al objetivo, que sería una billetera que alguien arrojara cerca.

Mientras tanto, algo estaba sucediendo que era indicativo de una lucha continua en este paraíso de hackers. David Silver estaba recibiendo muchas críticas. Las críticas venían de los némesis de la ética hacker: los teóricos de la IA y los estudiantes de posgrado en el octavo piso. Estas eran personas que no necesariamente veían el proceso de computación como un fin alegre en sí mismo: estaban más preocupados por obtener títulos, ganar reconocimiento profesional y, ejem, el avance de la ciencia informática. Consideraban que el hackerismo no era científico. Siempre exigían que los hackers salieran de la máquina para poder trabajar en sus “programas oficialmente aprobados”, y les horrorizaban los usos aparentemente frívolos que los hackers hacían de la computadora. Todos los estudiantes de posgrado estaban en medio de tesis y disertaciones académicas y científicas que pontificaban sobre la dificultad de hacer el tipo de cosas que David Silver estaba intentando. No considerarían ningún tipo de experimento de visión por computadora sin mucha más planificación, una revisión completa de experimentos anteriores, una arquitectura cuidadosa y una configuración que incluyera cubos blancos puros sobre terciopelo negro en una habitación prístina y sin polvo. Estaban furiosos porque el valioso tiempo del PDP-6 se estaba utilizando para este… ¡juguete! Por un adolescente inexperto, jugando con el PDP-6 como si fuera su kart personal.

Mientras los estudiantes de posgrado se quejaban de que David Silver nunca iba a llegar a nada, de que David Silver no estaba haciendo una IA adecuada y de que David Silver nunca iba a entender cosas como la teoría de funciones recursivas, David Silver seguía adelante con su insecto y su PDP-6. Alguien arrojó una cartera al suelo sucio y sucio y el insecto se deslizó hacia delante, quince centímetros por segundo, se movió a la derecha, se detuvo, siguió adelante. Y el estúpido bicho siguió lanzándose hacia delante, a la derecha o a la izquierda hasta que llegó a la cartera, y luego se abalanzó hasta que la cartera quedó sólidamente entre sus "astas" (que parecían perchas de camisa dobladas). Y luego el insecto empujó la cartera hacia su "bolígrafo" designado. Misión cumplida.

Los estudiantes de posgrado se volvieron completamente locos. Intentaron que echaran a Silver. Afirmaron que había consideraciones de seguro que surgían de la presencia de un chico de catorce años en el laboratorio a altas horas de la noche. Minsky tuvo que defender al chico. “En cierto modo los volvía locos”, reflexionó Silver más tarde, “porque este chico se quedaba haciendo tonterías durante unas semanas y la computadora comenzaba a hacer lo que estaban haciendo y que era realmente difícil, y ellos tenían dificultades y sabían que nunca resolverían por completo [el problema] y no podían implementarlo en el mundo real. Y de repente sucedió y los cabreé. Estaban teorizando todas estas cosas y yo me arremangaba y lo hacía... encuentras mucho de eso en el hacking en general. No lo abordaba desde un punto de vista teórico o de ingeniería, sino desde un

punto de vista divertido. Hagamos que este robot se mueva de una manera divertida e interesante. Y así, las cosas que construí y los programas que escribí realmente hicieron algo. Y en muchos casos, hicieron exactamente lo que estos estudiantes de posgrado estaban tratando de hacer".

Al final, los estudiantes de posgrado se calmaron con Silver. Pero el cisma fue constante. Los estudiantes de posgrado veían a los hackers como técnicos necesarios pero infantiles. Los hackers pensaban que los estudiantes de posgrado eran unos ignorantes con los pulgares en el culo que se sentaban en el octavo piso a teorizar a ciegas sobre cómo era la máquina. No sabrían qué era Lo Correcto ni aunque les cayera encima. Era una imagen ofensiva, esos incompetentes trabajando en Programas Sancionados Oficialmente que serían los temas de tesis y luego descartados (a diferencia de los programas de hackers, que se usaban y mejoraban). Algunos de ellos habían ganado sus sanciones engañando a profesores que no sabían casi nada sobre las máquinas. Los hackers veían a esa gente "descontrolarse" en el PDP-6 y lamentaban el desperdicio de tiempo de máquina perfectamente bueno.

Uno de estos estudiantes de posgrado, en particular, volvía locos a los hackers: cometía ciertos errores en sus programas que invariablemente hacían que la máquina intentara ejecutar instrucciones defectuosas, los llamados "códigos de operación no utilizados". Hacía esto durante horas y días seguidos. La máquina tenía una forma de lidiar con un código de operación no utilizado: lo almacenaba en un lugar determinado y, suponiendo que uno quisiera definir un nuevo código de operación, se preparaba para volver a él más tarde. Si uno no quería redefinir esa instrucción ilegal y procedía sin saber lo que había hecho, el programa entraba en un bucle, en cuyo momento uno lo detenía, revisaba el código y se daba cuenta de lo que había hecho mal. Pero este estudiante, al que llamaremos Fubar en lugar de su nombre, que había olvidado hace mucho tiempo, nunca pudo entenderlo y siguió introduciendo las instrucciones ilegales. Lo que hacía que la máquina entrara en un bucle desenfrenado, ejecutando constantemente instrucciones que no existían, esperando a que Fubar la detuviera. Fubar se quedaba sentado allí y miraba. Cuando recibía una copia impresa de su programa, la miraba fijamente. Más tarde, tal vez, cuando tenía la copia impresa en casa, se daba cuenta de su error y volvía a ejecutar el programa de nuevo. Entonces cometía el mismo error. Y los hackers se enfurecieron porque al llevarse a casa la copia impresa y arreglarla allí todo el tiempo, estaba desperdiciando el PDP-6, haciendo un procesamiento por lotes al estilo IBM en lugar de una programación interactiva. Era el equivalente al pecado capital.

Así que un día Nelson se metió en la computadora e hizo un hack que respondería a ese error en particular de una manera diferente. La gente se aseguró de quedarse cerca la próxima vez que Fubar se registró para la máquina. Se sentó en la consola, se tomó su habitual e interminable tiempo para ponerse en marcha, y efectivamente, al cabo de media hora, cometió el mismo estúpido error. Sólo que esta vez, en la pantalla de visualización, vio que el programa no estaba en bucle, sino que mostraba la parte de su código que había salido mal. Justo en el medio, señalando la instrucción ilegal que había introducido, había una enorme flecha brillante y fosforescente. Y parpadeando en la pantalla estaba la leyenda: "Fubar, ¡has perdido otra vez!".

Fubar no respondió amablemente. Se lamentó porque su programa había sido vandalizado. Estaba tan indignado que ignoró por completo la información que el hack de Nelson le había dado sobre lo que estaba haciendo mal y lo que podría hacer para solucionarlo. No estaba, como los hackers habían esperado de alguna manera, agradecido de que se hubiera instalado esta maravillosa

característica para ayudarlo a encontrar el error de sus métodos. La brillantez del hack se había desperdiciado en él.

• • • • • • •

Los hackers tenían una palabra para describir a esos estudiantes de posgrado. Era la misma palabra que usaban para describir a casi cualquiera que pretendiera saber algo sobre computadoras y no pudiera respaldarlo con la experiencia de un hacker. La palabra era "lose" (“perdedor”). Los hackers eran "ganadores" (“winners”). Era una distinción binaria: las personas alrededor del laboratorio de IA eran una cosa o la otra. El único criterio era la capacidad de hackear. Tan intensa era la búsqueda de mejorar el mundo mediante la comprensión y la construcción de sistemas que casi todos los demás rasgos humanos eran ignorados. Podías tener catorce años y ser disléxico, y ser un ganador. O podías ser brillante, sensible y estar dispuesto a aprender, y aun así ser considerado un perdedor.

Para un recién llegado, el noveno piso era un palacio de la pasión científica intimidante y aparentemente impenetrable. El simple hecho de estar cerca de personas como Greenblatt, Gosper o Nelson podía ponerte la piel de gallina. Parecían las personas más inteligentes del mundo. Y como sólo una persona a la vez podía utilizar el PDP-6, hacía falta mucho coraje para sentarse y aprender cosas de forma interactiva. Aun así, cualquiera que tuviera el espíritu de hacker en él estaría tan motivado a calcular que dejaría de lado las dudas sobre sí mismo y comenzaría a escribir programas.

Tom Knight, que subió al noveno piso como un estudiante de primer año de diecisiete años sorprendentemente alto y delgado en 1965, pasó por ese proceso y finalmente ganó el estatus de ganador. Para lograrlo, recordó más tarde, “tienes que sumergirte en esa cultura. Largas noches mirando por encima del hombro de personas que estaban haciendo cosas interesantes que tú no entendías”. Lo que lo mantuvo en marcha fue su fascinación por la máquina; cómo te permitía construir sistemas complicados completamente bajo tu control. En ese sentido, Knight se refirió más tarde a que en el caso de los ordenadores, se tenía el mismo tipo de control que un dictador sobre un sistema político. Pero Knight también creía que eran un medio artístico infinitamente flexible, en el que uno podía expresarse creando su propio pequeño universo. Knight explicó más tarde: “Aquí está este objeto al que puedes decirle qué hacer y, sin que le hagan preguntas, hace lo que le dices. Hay muy pocas instituciones en las que una persona de dieciocho años pueda conseguir que eso le suceda”.

Personas como Knight y Silver hackeaban tan intensamente y tan bien que se convirtieron en ganadores. Otros se enfrentaron a un largo camino cuesta arriba, porque una vez que los hackers sentían que uno era un obstáculo para la mejora general del sistema en su conjunto, era un perdedor en el peor sentido y debía ser tratado con indiferencia o decirle que se fuera directamente.

Para algunos, eso parecía cruel. Un hacker sensible llamado Brian Harvey estaba particularmente molesto por la drástica aplicación de la norma. Harvey pasó la prueba con éxito. Mientras trabajaba con la computadora descubrió algunos errores en el editor TECO y, cuando los señaló, la gente le dijo: “Bien, ahora ve a arreglarlos”. Lo hizo, se dio cuenta de que el proceso de depuración era más divertido que usar un programa que ya había depurado y se puso a buscar más errores para arreglar. Un día, mientras estaba hackeando TECO, Greenblatt se paró detrás de él, acariciándole la barbilla mientras Harvey tecleaba algo de código, y dijo: “Supongo que deberíamos comenzar a pagarte”. Así era como te contrataban en el laboratorio. Solo contrataban a los ganadores.

Pero a Harvey no le gustaba que a otras personas se las señalara como perdedoras, se las tratara como parias simplemente porque no eran brillantes. Harvey pensaba que Marvin Minsky tenía mucho que ver con promulgar esa actitud. (Minsky insistió más tarde en que lo único que hizo fue permitir que los hackers manejaran las cosas por sí mismos: “el sistema era abierto y literalmente alentaba a la gente a probarlo, y si eran dañinos o incompetentes, se les animaba a irse”). Harvey reconoció que, si bien por un lado el laboratorio de IA, alimentado por la ética hacker, era “un gran jardín intelectual”, por otro lado tenía el defecto de que no importaba tanto quién eras sino qué tipo de hacker eras.

Algunas personas cayeron en la trampa de esforzarse tanto por ser un ganador en la máquina que fueron juzgados instantáneamente como perdedores: por ejemplo, Gerry Sussman, que llegó al MIT como un arrogante adolescente de diecisiete años. Habiendo sido un adolescente adicto a la electrónica y un fanático de las computadoras en la escuela secundaria, lo primero que hizo cuando llegó al MIT fue buscar una computadora. Alguien le señaló Tech Square. Le preguntó a una persona que parecía pertenecer allí si podía jugar con la computadora. Richard Greenblatt le dijo, adelante, juega con ella.

De modo que Sussman empezó a trabajar en un programa. Poco después, un tipo calvo de aspecto extraño se acercó a él. Sussman pensó que lo iba a echar, pero en lugar de eso, el hombre se sentó y le preguntó: "Oye, ¿qué estás haciendo?". Sussman habló sobre su programa con el hombre, Marvin Minsky. En un momento de la conversación, Sussman le dijo a Minsky que estaba usando cierta técnica de aleatorización en su programa porque no quería que la máquina tuviera nociones preconcebidas. Minsky dijo: "Bueno, las tiene, es solo que no sabes qué son". Fue lo más profundo que Gerry Sussman había escuchado nunca. Y Minsky continuó, diciéndole que el mundo está construido de cierta manera, y que lo más importante que podemos hacer con el mundo es evitar la aleatoriedad y descubrir formas de planificar las cosas. Una sabiduría como esta tiene su efecto en los estudiantes de primer año de diecisiete años, y desde entonces Sussman quedó enganchado.

Pero empezó con mal pie con los hackers. Trató de compensar su inseguridad con una bravuconería excesiva, y todo el mundo se dio cuenta. También era, según muchos, terriblemente torpe, casi acabó aplastado en un combate con el brazo robótico (que le costaba muchísimo controlar) y una vez aplastó accidentalmente una pelota de ping-pong de marca especial importada que Gosper había llevado al laboratorio. En otra ocasión, durante una aventura de la Sociedad de Cableado Informático de Medianoche, a Sussman se le metió una gota de soldadura en el ojo. Estaba perdiendo por todos lados.

Quizás para cultivar una imagen elegante, Sussman fumaba en pipa, algo totalmente incorrecto en el noveno piso, donde el humo era apestoso, y un día los hackers lograron reemplazar parte de su tabaco con bandas elásticas cortadas en pedazos del mismo color aproximadamente.

Se convirtió unilateralmente en aprendiz de Gosper, el más profundo verbalmente de los hackers. Gosper tal vez no pensara que Sussman fuera un gran ganador en ese momento, pero le encantaba tener público y toleraba la arrogancia equivocada de Sussman. A veces, los comentarios irónicos del gurú hacían que Sussman se mareara, como cuando Gosper comentó despreocupadamente que "Bueno, los datos son solo un tipo de programación tonta". Para Sussman, eso respondía a la pregunta de la existencia eterna: "¿Qué eres tú?" Somos datos, piezas de un programa informático cósmico que es el universo. Al observar los programas de Gosper, Sussman adivinó que dicha filosofía estaba como incrustada en el código. Sussman explicó más tarde que “Gosper de alguna

manera imaginó que el mundo estaba hecho de todas estas pequeñas piezas, cada una de las cuales es una pequeña máquina que es un pequeño estado local independiente. Y [cada estado] hablaría con sus vecinos".

Al observar los programas de Gosper, Sussman se dio cuenta de una suposición importante del hackerismo: todos los programas informáticos serios son expresiones de un individuo. "Es solo incidental que las computadoras ejecuten programas", explicaría más tarde Sussman. "Lo importante de un programa es que es algo que puedes mostrar a las personas, y ellas pueden leerlo y pueden aprender algo de él. Lleva información. Es una parte de tu mente que puedes escribir y dar a otra persona como un libro". Sussman aprendió a leer programas con la misma sensibilidad con la que un aficionado a la literatura leería un poema. Hay programas divertidos que contienen chistes, hay programas emocionantes que hacen lo correcto y hay programas tristes que hacen intentos valientes pero no funcionan del todo.

Son cosas importantes que hay que saber, pero no necesariamente te convirtieron en un ganador. Fue el hacking lo que lo convirtió en un ganador para Sussman. Se mantuvo firme, estuvo mucho tiempo con Gosper, suavizó su actitud de sabelotodo y, sobre todo, se convirtió en un programador impresionante. Era el raro perdedor que finalmente dio la vuelta a la situación y se convirtió en un ganador. Más tarde escribió un programa muy complicado y muy publicitado en el que la computadora movería bloques con un brazo robótico y, mediante un proceso muy parecido a la depuración, el programa averiguaría por sí mismo qué bloques tendría que mover para llegar al que se le había solicitado. Fue un avance significativo para la inteligencia artificial y, a partir de entonces, Sussman se hizo conocido más como un científico, un planificador. Llamó a su famoso programa HACKER.

Una cosa que ayudó a Sussman en su cambio de perdedor a ganador fue su sentido de lo que era Lo Correcto. Los mayores perdedores de todos, a ojos de los hackers, fueron aquellos que carecían tanto de esa capacidad que eran incapaces de darse cuenta de cuál era la verdadera mejor máquina, o el verdadero mejor lenguaje informático, o la verdadera mejor manera de utilizar un ordenador. Y ningún sistema de utilización de un ordenador se ganó tanto el desprecio de los hackers como los sistemas de tiempo compartido que, como eran una parte importante del Proyecto MAC, también estaban basados en el noveno piso de Tech Square. El primero, que estaba en funcionamiento desde mediados de los años sesenta, fue el Compatible Time-sharing System (CTSS). El otro, que tardó mucho en prepararse y era muy costoso, se llamaba Multics y era tan ofensivo que su mera existencia era un escándalo.

A diferencia del mosaico de programas de sistemas en constante mejora que funcionaban en el PDP-6, el CTSS había sido escrito por un solo hombre, el profesor del MIT F.J. Corbató. Había sido un trabajo virtuoso en muchos aspectos, todo cuidadosamente codificado y listo para funcionar en el IBM 7094, que soportaría una serie de terminales que se utilizarían simultáneamente. Pero para los hackers, el CTSS representaba la burocracia y el IBMismo. "Una de las cosas realmente divertidas de los ordenadores es que tienes control sobre ellos", explicaría más tarde Tom Knight, enemigo del CTSS. "Cuando tienes una burocracia alrededor de un ordenador, ya no tienes control sobre él. El CTSS era un sistema 'serio'. La gente tenía que ir a buscar cuentas y prestar atención a la seguridad. Era una burocracia benigna, pero al fin y al cabo una burocracia, llena de gente que estaba allí de nueve a cinco. Si había alguna razón para que quisieras cambiar el comportamiento del sistema, la forma en que funcionaba, o desarrollar un programa que podría haber funcionado solo a veces, o podría tener algún peligro de hacer que el sistema se estropeara, eso no se fomentaba [en el CTSS].

Quieres un entorno en el que cometer esos errores no sea algo por lo que te castiguen, sino un entorno en el que la gente diga: 'Ups, cometiste un error'". En otras palabras, el CTSS desalentaba el hackeo. Si a esto añadimos el hecho de que funcionaba en una máquina IBM de dos millones de dólares que los hackers consideraban muy inferior a su PDP-6, el resultado era un sistema perdedor. Nadie les pedía a los hackers que utilizaran el CTSS, pero estaba ahí, y a veces sólo había que hacer algo de hacking con lo que había disponible. Cuando un hacker intentaba utilizarlo y aparecía un mensaje en la pantalla diciendo que no se podía iniciar sesión sin la contraseña adecuada, se veía obligado a tomar represalias. Porque para los hackers, las contraseñas eran incluso más odiosas que las puertas cerradas. ¿Qué podía ser peor que alguien te dijera que no estabas autorizado a utilizar su ordenador?

Al final resultó que los hackers aprendieron a utilizar el sistema CTSS tan bien que podían eludir los requisitos de contraseña. Una vez que estaban en el sistema, se lo restregaban un poco a los administradores dejando mensajes, equivalentes de alta tecnología a "Kilroy estuvo aquí". A veces incluso conseguían que el ordenador imprimiera una lista de todas las contraseñas actuales y dejaban la copia impresa debajo de la puerta de un administrador. Greenblatt recuerda que la gente del Proyecto MAC-CTSS no vio con buenos ojos eso e insertó un memorando MAC oficial, que aparecía cuando iniciabas sesión, básicamente diciendo, una contraseña es tu santidad, y sólo la forma más baja de ser humano violaría una contraseña. Tom Knight entró en el sistema y cambió el encabezado de ese memo de MAC a HAC.

Pero por muy malo que fuera CTSS, los hackers pensaron que Multics era peor. Multics era el nombre del sistema de tiempo compartido enormemente caro para las masas que se estaba construyendo y depurando en el noveno piso. Aunque estaba diseñado para usuarios generales, los hackers evaluaban la estructura de cualquier sistema desde una perspectiva muy personal, especialmente un sistema creado en el mismo piso del edificio en el que hackeaban. Así que MULTICS fue un gran tema de conversación entre los hackers.

Originalmente, Multics se hizo en conjunto con General Electric; luego intervino Honeywell. Hubo todo tipo de problemas con él. Tan pronto como los hackers escucharon que el sistema funcionaría en terminales de teletipo Modelo 33 en lugar de pantallas CRT rápidas e interactivas, supieron que el sistema era un fracaso total. El hecho de que el sistema estuviera escrito en un lenguaje de programación creado por IBM llamado PL/I en lugar de un elegante lenguaje de máquina era espantoso. Cuando el sistema funcionó por primera vez, era increíblemente lento. Era tan lento que los hackers concluyeron que todo el sistema debía tener daño cerebral, un término usado tan a menudo para describir a Multics que "daño cerebral" se convirtió en un término peyorativo estándar en la jerga hacker.

Pero lo peor de Multics era la fuerte seguridad y el sistema de cobrar al usuario por el tiempo. Multics adoptó la actitud de que el usuario pagaba hasta el último centavo; cobraba una parte por la memoria que usaba, otra por el espacio en disco, otra por el tiempo. Mientras tanto, los planificadores de Multics, desde el punto de vista de los hackers, hacían proclamas sobre cómo esta era la única forma en que las utilidades podían funcionar. El sistema cambió totalmente la ética de los hackers: en lugar de alentar a pasar más tiempo frente al ordenador (lo único bueno del tiempo compartido en lo que a la mayoría de los hackers se refiere), instaba a pasar menos tiempo y a utilizar menos las funciones del ordenador una vez que se estaba conectado. La filosofía de Multics era un desastre.

Los hackers plagaron el sistema Multics con trucos y fallos. Era casi una obligación hacerlo. Como Minsky diría más tarde: “Había gente haciendo proyectos que a otras personas no les gustaban y les gastaban todo tipo de bromas para que fuera imposible trabajar con ellos... Creo que [los hackers] ayudaron al progreso socavando a los profesores con planes estúpidos”.

A la luz de las tendencias guerrilleras de los hackers, los planificadores a cargo del laboratorio de IA tuvieron que andarse con mucho cuidado con las sugerencias que afectarían al entorno de los hackers. Y alrededor de 1967, los planificadores querían un cambio enorme. Querían convertir el querido PDP-6 de los hackers en una máquina de tiempo compartido.

En ese momento, Minsky había delegado muchas de sus funciones de liderazgo del laboratorio de IA en su amigo Ed Fredkin, el jefe de Nelson en Triple-I, que estaba dejando su trabajo a tiempo completo para pasar a ser profesor en el MIT (Fredkin sería uno de los profesores titulares más jóvenes de la facultad y el único profesor titular sin título). Fredkin, un maestro de la programación, ya tenía una relación estrecha con los hackers. Apreciaba la forma en que la actitud de laissez-faire permitía a los hackers ser deslumbrantemente productivos, pero pensaba que a veces los hackers podían beneficiarse de una dirección de arriba hacia abajo. Uno de sus primeros intentos de organizar un enfoque de “ola humana” para abordar un problema de robótica, asignando él mismo a los hackers partes específicas del problema, había fracasado estrepitosamente. “Todos pensaban que estaba loco”, recordó Fredkin más tarde. Al final aceptó el hecho de que la mejor manera de conseguir que los hackers hicieran cosas era sugerírselas y esperar que se interesaran lo suficiente. Entonces se obtendría una producción inaudita en la industria o en el mundo académico.

Minsky y Fredkin consideraban que el tiempo compartido era algo esencial. Entre los hackers y los usuarios autorizados oficialmente, el PDP-6 tenía una demanda constante; la gente se sentía frustrada por las largas esperas para acceder. Pero los hackers no consideraban aceptable el tiempo compartido. Señalaban a CTSS, Multics e incluso al sistema más amable de Jack Dennis en el PDP-1 como ejemplos del acceso más lento y menos potente con el que uno se vería obligado a quedarse atascado cuando compartiera el ordenador con otros que lo usaran al mismo tiempo.

Observaron que ciertos programas grandes no podían ejecutarse en absoluto con el tiempo compartido. Uno de ellos era un programa gigantesco en el que Peter Samson había estado trabajando. Era una especie de consecuencia de uno de sus primeros hackeos en el TX-0, un programa que, si escribías los nombres de dos estaciones de metro en la MTA, te indicaba las líneas de metro adecuadas que debías tomar y dónde hacer los cambios de una a otra. Ahora, Samson estaba abordando todo el sistema de metro de Nueva York... Su intención era poner todo el sistema en la memoria de la computadora y el horario completo de sus trenes en un disco de datos al que la computadora pudiera acceder. Un día ejecutó el programa para determinar una ruta por la que una persona pudiera recorrer todo el sistema de metro con una sola ficha. Recibió cierta atención de los medios y luego alguien sugirió que vieran si podían usar la computadora para hacerlo realmente, batir un récord previamente establecido por un estudiante de Harvard al viajar realmente a cada parada del sistema de metro de Nueva York.

Después de meses de hackeo, Samson ideó un plan, y un día dos hackers lograron el objetivo. Se instaló un teletipo en el Club de Antiguos Alumnos del MIT en Manhattan, conectado al PDP-6. Se apostaron unas dos docenas de mensajeros a lo largo de la ruta, que periódicamente se metían en los teléfonos públicos, actualizando constantemente la información de los horarios, llamando a los trenes que llegaban tarde, informando de los retrasos y anotando las conexiones perdidas. los

hackers del teletipo introducían la información y, de vuelta en Cambridge, el PDP-6 calculaba los cambios en la ruta. A medida que los viajeros pasaban por cada estación, Samson la marcaba en un mapa de la sala de guerra. La idea de estos locos con el pelo al rape (en marcado contraste con los manifestantes de pelo largo que hacían noticia en otros tipos de actividades) capturó la imaginación de los medios durante un día, y El Gran Hackeo del Metro se destacó como uno de los usos memorables del PDP-6.

Subrayó algo que Greenblatt, Gosper y el resto consideraban esencial: la magia que sólo podía surgir de programas que utilizaran todo el ordenador. los hackers trabajaban en el PDP-6, uno por uno, como si fuera su propio ordenador personal. A menudo ejecutaban programas de visualización que se ejecutaban en "tiempo real" y requerían que el ordenador refrescara constantemente la pantalla; el tiempo compartido haría que los programas de visualización se ejecutaran más lentamente. Y los hackers se habían acostumbrado a los pequeños adornos que surgían del control total del PDP-6, como poder seguir un programa mediante las luces intermitentes (que indicaban qué registros de la máquina se estaban activando). Esas ventajas desaparecerían con el tiempo compartido.

Sin embargo, en el fondo, la cuestión del tiempo compartido era una cuestión estética. La idea misma de que no se pudiera controlar toda la máquina era inquietante. Incluso si el sistema de tiempo compartido permitiera que la máquina respondiera a uno exactamente de la misma manera que lo hacía en modo de usuario único, uno sabría que no era todo suyo. Sería como intentar hacerle el amor a tu mujer sabiendo que ella al mismo tiempo está haciéndole el amor a otras seis personas.

La obstinación de los hackers en este asunto ilustraba su compromiso con la calidad de la informática; no estaban dispuestos a comprometerse utilizando un sistema inferior que serviría a más gente y tal vez difundiría el evangelio del hacking. En su opinión, el hacking sería mejor si se utilizara el mejor sistema posible. No un sistema de tiempo compartido.

Fredkin se enfrentaba a una lucha política cuesta arriba. Su estrategia era dar la vuelta al más vehemente del bando contrario al tiempo compartido: Greenblatt. Había cierto afecto entre ellos. Fredkin era la única persona en el noveno piso que llamaba a Greenblatt "Ricky". Así que lo cortejó. Lo engatusó. Le dijo a Greenblatt que la potencia del PDP-6 mejoraría con una nueva pieza de hardware, que ampliaría su memoria a un tamaño mayor que el de cualquier ordenador del mundo. Prometió que el sistema de tiempo compartido sería mejor que cualquiera hasta la fecha... y que los hackers lo controlarían. Trabajó en Greenblatt durante semanas y, finalmente, Ricky Greenblatt aceptó que se implementara el tiempo compartido en el PDP-6.

Poco después, Fredkin estaba en su oficina cuando Bill Gosper entró, liderando a varios hackers. Se pusieron en fila frente al escritorio de Fredkin y le dirigieron una mirada gélida colectiva.

"¿Qué pasa?", preguntó Fredkin.

Se quedaron mirándolo un rato más. Finalmente, hablaron.

"Nos gustaría saber qué le han hecho a Greenblatt", dijeron. "Tenemos razones para creer que lo han hipnotizado".

Gosper, en particular, tuvo dificultades para aceptar el control conjunto del PDP-6. Su comportamiento le recordó a Fredkin a Rourke, el arquitecto de El manantial de Ayn Rand, que diseñó un hermoso edificio; cuando los superiores de Rourke tomaron el control del diseño y comprometieron su belleza, Rourke hizo estallar el edificio. Fredkin recordó más tarde que Gosper

le dijo que si se implementaba el tiempo compartido en el PDP-6, Gosper se vería obligado a demoler físicamente la máquina. “Al igual que Rourke”, recordó Fredkin más tarde. “Él pensaba que si se iba a hacer algo tan terrible, habría que destruirla. Y yo entendía ese sentimiento. Así que elaboré un compromiso”. El compromiso permitía que la máquina funcionara a altas horas de la noche en modo de usuario único para que los hackers pudieran ejecutar programas de pantalla gigantes y tener el PDP-6 bajo su total control.

Todo el experimento de tiempo compartido no salió nada mal. La razón fue que se creó un nuevo sistema especial de tiempo compartido, un sistema que tenía la ética hacker en su misma alma.

• • • • • • •

El núcleo del sistema fue escrito por Greenblatt y Nelson, en semanas de hacking hardcore. Una vez que se terminó parte del software, Tom Knight y otros comenzaron los ajustes necesarios en el PDP-6 y la nueva memoria adicional: un gabinete grande con el tamaño de dos lavadoras del tamaño de una lavandería automática, apodado Moby Memory. Aunque la administración aprobó que los hackers trabajaran en el sistema, Greenblatt y los revisores ejercieron autoridad total sobre cómo resultaría el sistema. Una indicación de cómo este sistema se diferenciaba de los demás (como el Sistema de Tiempo Compartido Compatible) fue el nombre que Tom Knight le dio al programa hacker: el Sistema de Tiempo Compartido Incompatible (ITS).

El título era particularmente irónico porque, en términos de compatibilidad con otros sistemas y programas, el ITS era mucho más compatible que el CTSS. Fiel a la Ética Hacker, el ITS podía vincularse fácilmente con otras cosas, de esa manera podría extenderse infinitamente para que los usuarios pudieran explorar el mundo de manera más efectiva. Como en cualquier sistema de tiempo compartido, varios usuarios podrían ejecutar programas en el ITS al mismo tiempo. Pero en el ITS, un usuario también podría ejecutar varios programas a la vez. El ITS también permitía un uso considerable de las pantallas y tenía lo que para la época era un sistema de edición muy avanzado que usaba la pantalla completa ("años antes que el resto del mundo", se jactó más tarde Greenblatt). Como los hackers querían que la máquina funcionara tan rápido como lo hubiera hecho si no hubiera sido compartida, Greenblatt y Nelson escribieron un código en lenguaje de máquina que permitía un control sin precedentes en un sistema de tiempo compartido.

Había una encarnación aún más llamativa de la ética hacker dentro de ITS. A diferencia de casi cualquier otro sistema de tiempo compartido, ITS no utilizaba contraseñas. De hecho, estaba diseñado para permitir a los hackers el máximo acceso a los archivos de cualquier usuario. La vieja práctica de tener cintas de papel en un cajón, una biblioteca de programas colectiva donde la gente usaba y mejoraba sus programas, estaba incorporada en ITS; cada usuario podía abrir un conjunto de archivos personales, almacenados en un disco. La arquitectura abierta de ITS alentaba a los usuarios a revisar estos archivos, ver en qué ingeniosos hacks estaban trabajando otras personas, buscar errores en los programas y corregirlos. Si querías una rutina para calcular funciones seno, por ejemplo, podías buscar en los archivos de Gosper y encontrar su hack de seno de diez instrucciones. Se podían recorrer los programas de los hackers maestros, buscando ideas, admirando el código. La idea era que los programas de ordenador no pertenecían a individuos, sino al mundo de los usuarios.

ITS también conservaba el sentimiento de comunidad que tenían los hackers cuando había un solo usuario en la máquina y la gente podía apiñarse a su alrededor para verlo codificar. Mediante una inteligente conmutación de barras cruzadas, cualquier usuario de ITS no sólo podía escribir un

comando para averiguar quién más estaba en el sistema, sino que podía cambiarse a la terminal de cualquier usuario que quisiera supervisar. Incluso se podía hackear en conjunción con otro usuario: por ejemplo, Knight podía iniciar sesión, averiguar que Gosper estaba en uno de los otros puertos y llamar a su programa; luego podía escribir líneas de código en el programa que Gosper estaba hackeando.

Esta característica se podía utilizar de muchas maneras. Más adelante, después de que Knight hubiera construido algunos terminales gráficos sofisticados, un usuario podía estar trabajando en un programa y de repente en la pantalla aparecía este bicho de seis patas... Se arrastraba por la pantalla y tal vez comenzaba a masticar el código, esparciendo pequeñas migajas de fósforo por todas partes. En otra terminal, histérico y con una risa aguda, estaba el hacker que te decía, de esa manera inescrutable, que tu programa tenía errores. Pero aunque cualquier usuario tenía el poder no sólo de hacer ese tipo de cosas, sino de entrar en tus archivos y borrar (“reap”, como lo llamaban) tus programas y notas valiosas hackeadas, ese tipo de cosas no se hacían. Había honor entre los hackers en ITS.

La fe que ITS tenía en los usuarios se demostró mejor en su manejo del problema de los colapsos intencionales del sistema. Anteriormente, un rito de iniciación de los hackers era entrar en un sistema de tiempo compartido y causar tal caos digital (tal vez sobrecargando los registros con cálculos en bucle) que el sistema “colapsaba”, se quedaba completamente muerto. Después de un tiempo, un hacker dejaba ese modo destructivo, pero sucedía con la suficiente frecuencia como para ser un problema considerable para las personas que tenían que trabajar en el sistema. Cuanto más protecciones tuviera el sistema contra esto, mayor sería el desafío para un hacker cualquiera que lo pusiera de rodillas. Multics, por ejemplo, requería un hackeo verdaderamente no trivial antes de que fracasara. Así que siempre había programadores machos que demostraban su valía haciendo que Multics se bloqueara.

ITS, en cambio, tenía un comando cuya función específica era bloquear el sistema. Todo lo que había que hacer era escribir KILL SYSTEM y el PDP-6 se paralizaba. La idea era quitarle toda la diversión a bloquear el sistema haciendo que fuera trivial hacerlo. En raras ocasiones, algún perdedor miraba los comandos disponibles y decía: "¿Me pregunto qué hace KILL?" y hacía que el sistema se cayera, pero en general ITS demostró que la mejor seguridad era no tener seguridad en absoluto.

Por supuesto, tan pronto como ITS se instaló en el PDP-6 hubo una oleada de depuración que, en cierto sentido, se prolongó durante más de una década. Greenblatt fue el más destacado de los que pasaron días enteros "hackeando ITS": buscando errores, agregando nuevas características, haciendo que secciones del sistema funcionen más rápido... trabajando tanto en el que el ambiente ITS se convirtió, en efecto, en un hogar para los hackers de sistemas.

En el mundo que era el laboratorio de IA, el papel del hacker de sistemas era central. La ética hacker permitía a cualquiera trabajar en ITS, pero las consecuencias públicas del hackeo de sistemas arrojaban una dura luz sobre la calidad de tu trabajo: si intentabas mejorar el ensamblador MIDAS o el depurador ITS-DDT y cometías un error horrible, los programas de todos se iban a bloquear y la gente iba a descubrir quién era el perdedor responsable. Por otro lado, no había vocación más alta en el hackerismo que el hackeo de sistemas de calidad.

Los planificadores no consideraban el hackeo de sistemas con la misma estima. Los planificadores estaban preocupados por las aplicaciones: usar computadoras para ir más allá de la computación,

para crear conceptos y herramientas útiles para beneficiar a la humanidad. Para los hackers, el sistema era un fin en sí mismo. Después de todo, la mayoría de los hackers habían estado fascinados por los sistemas desde la primera infancia. Habían dejado de lado casi todo lo demás en la vida una vez que reconocieron que la herramienta definitiva para crear sistemas era el ordenador: no sólo se podía utilizar para crear un sistema increíblemente complicado, a la vez bizantino y elegantemente eficiente, sino que además, con un sistema operativo "Moby" como ITS, ese mismo ordenador podía ser en realidad el sistema. Y la belleza de ITS era que se abría, facilitaba la escritura de programas que encajaran en él, pedía nuevas funciones y adornos. ITS era la sala de estar de los hackers, y todo el mundo podía hacer lo que pudiera para sentirse cómodo; para encontrar y decorar su propio pequeño nicho. ITS era el sistema perfecto para construir... ¡sistemas!

Era un bucle lógico en espiral sin fin. A medida que la gente utilizaba ITS, podía admirar esta o aquella característica, pero lo más probable era que pensara en formas de mejorarlo. Esto era natural, porque un corolario importante del hackerismo establece que ningún sistema o programa está nunca completo. Siempre se puede mejorar. Los sistemas son creaciones orgánicas y vivas: si la gente deja de trabajar en ellos y mejorarlos, mueren.

Cuando completabas un programa de sistemas, ya fuera un gran esfuerzo como un ensamblador o un depurador o algo rápido y (esperabas) elegante, como un multiplexor de salida de interfaz, estabas creando simultáneamente una herramienta, revelando una creación y dando forma a algo para mejorar el nivel de tu propio hackeo futuro. Era un proceso particularmente circular, casi espiritual, en el que el programador de sistemas era un usuario habitual del sistema que estaba mejorando. Muchos programas de sistemas virtuosos surgieron de remedios a obstáculos molestos que los hackers sentían que les impedían programar de manera óptima. (La programación óptima real, por supuesto, solo se podía lograr cuando se eliminaban todos los obstáculos entre tú y la computadora pura, un ideal que probablemente no se cumplirá hasta que los hackers se fusionen biológicamente de alguna manera con las computadoras). Los programas que escribieron los hackers de ITS los ayudaron a programar más fácilmente, hicieron que los programas se ejecutaran más rápido y permitieron que los programas se beneficiaran del poder que proviene de usar más de la máquina. Así, un hacker no sólo obtendría una enorme satisfacción al escribir un programa de sistemas brillante (una herramienta que todo el mundo usaría y admiraría), sino que, a partir de entonces, estaría mucho más avanzado en la creación del siguiente programa de sistemas.

Para citar un informe de progreso escrito por el hacker Don Eastlake cinco años después de que ITS se pusiera en funcionamiento por primera vez:

> El sistema ITS no es el resultado de una oleada humana ni de un esfuerzo de choque. El sistema se ha ido desarrollando de forma incremental casi de forma continua desde su inicio. Es cierto que los grandes sistemas nunca están "terminados". . . En general, se puede decir que el sistema ITS ha sido diseñado por el diseñador y por el usuario. El problema del diseño de software poco realista se reduce enormemente cuando el diseñador es el implementador. La facilidad del implementador para programar y el orgullo por el resultado aumentan cuando él, en un sentido esencial, es el diseñador. Es menos probable que las características resulten de poca utilidad si los usuarios son sus diseñadores y es menos probable que sean difíciles de usar si sus diseñadores son sus usuarios.

La prosa era densa, pero el punto era claro: ITS era la expresión más fuerte hasta ahora de la ética hacker. Muchos pensaban que debería ser un estándar nacional para los sistemas de tiempo

compartido en todas partes. Que todos los sistemas informáticos del país difundieran el evangelio, eliminando el odioso concepto de las contraseñas, instando a la práctica práctica sin restricciones de la depuración de sistemas y demostrando el poder sinérgico que surge del software compartido, donde los programas pertenecen no al autor sino a todos los usuarios de la máquina.

En 1968, las principales instituciones informáticas celebraron una reunión en la Universidad de Utah para elaborar un sistema estándar de tiempo compartido que se utilizaría en la última máquina de DEC, la PDP-10. La Ten sería muy similar a la PDP-6, y uno de los dos sistemas operativos en consideración era el Sistema de tiempo compartido incompatible de los hackers. El otro era TENEX, un sistema escrito por Bolt Beranek y Newman que aún no se había implementado. Greenblatt y Knight representaron al MIT en la conferencia y presentaron una imagen extraña: dos hackers que intentaban persuadir a las burocracias reunidas de una docena de grandes instituciones para que comprometieran millones de dólares de su equipo a un sistema que, para empezar, no tenía seguridad incorporada.

Fracasaron.

Knight diría más tarde que fue la ingenuidad política la que hizo que los hackers del MIT perdieran la batalla. Supuso que la solución estaba lista incluso antes de que se convocara la conferencia: un sistema basado en la Ética Hacker era una medida demasiado drástica para que la tomaran esas instituciones. Pero Greenblatt insistió más tarde en que "podríamos haber ganado si realmente hubiéramos querido". Pero "seguir adelante", como él lo expresó, era más importante. Simplemente no era una prioridad para Greenblatt difundir la Ética Hacker mucho más allá de los límites de Cambridge. Consideró mucho más importante centrarse en la sociedad de Tech Square, la utopía hacker que asombraría al mundo al aplicar la Ética Hacker para crear sistemas cada vez más perfectos.

Capítulo 7. LIFE

Más tarde llamarían a esta maravillosa existencia en el noveno piso de Tech Square la Edad de Oro del hacking. Pasaban el tiempo en la monótona sala de máquinas y en las desordenadas oficinas cercanas, reunidos alrededor de terminales donde filas y filas de caracteres verdes de código se desplazaban frente a ellos, marcando impresiones con lápices sacados de los bolsillos de las camisas y charlando en su peculiar jerga sobre este bucle infinito o aquella subrutina fallida; el grupo de monjes tecnológicos que poblaban el laboratorio estaba tan cerca del paraíso como nunca lo estarían. Un estilo de vida anarquista benévolo dedicado a la productividad y a la pasión por el PDP-6. El arte, la ciencia y el juego se habían fusionado en la actividad mágica de la programación, y cada hacker era un maestro omnipotente del flujo de información dentro de la máquina. La vida depurada en todo su esplendor.

Pero por mucho que los hackers intentaran vivir el sueño hacker sin la interferencia de los patéticamente deformados sistemas del "mundo real", no podían hacerlo. El fracaso de Greenblatt y Knight en convencer a los forasteros de la superioridad natural del Sistema Incompatible de Tiempo Compartido fue sólo un indicio de que la inmersión total de un pequeño grupo de personas en el hackerismo podría no producir un cambio a la escala masiva que todos los hackers suponían que era inevitable. Era cierto que, en la década transcurrida desde que el TX-0 fue entregado por primera vez al MIT, el público en general y, sin duda, los demás estudiantes del campus se habían vuelto más conscientes de las computadoras en general. Pero no consideraban a las computadoras con el mismo respeto y fascinación que los hackers. Y no consideraban necesariamente que las intenciones de los hackers fueran benignas e idealistas.

Por el contrario, muchos jóvenes de finales de los años 1960 veían a las computadoras como algo maligno, parte de una conspiración tecnológica en la que los ricos y poderosos usaban el poder de la computadora contra los pobres e impotentes. Esta actitud no se limitaba a los estudiantes que protestaban, entre otras cosas, por la ahora explosiva Guerra de Vietnam (un conflicto librado en parte por computadoras estadounidenses). Las máquinas que constituían el alma del hackerismo también eran detestadas por millones de ciudadanos comunes y patriotas que veían a las computadoras como un factor deshumanizador de la sociedad. Cada vez que llegaba a un hogar una factura incorrecta y los intentos del destinatario de corregirla terminaban en una ronda frustrante de llamadas, que generalmente conducían a una explicación de que “el ordenador lo había hecho” y que sólo un esfuerzo humano hercúleo podía borrar la mancha digital, el desprecio popular hacia las computadoras crecía. Los hackers, por supuesto, atribuían esos errores a la mentalidad burocrática, de procesamiento por lotes y con daño cerebral de IBM. ¿No entendía la gente que la Ética Hacker eliminaría esos abusos al alentar a la gente a arreglar errores como facturas de electricidad de mil dólares? Pero en la mente del público no había distinción entre los programadores de Hulking Giants y los habitantes del laboratorio de inteligencia artificial del elegante e interactivo PDP-6. Y en esa mentalidad pública todos los programadores informáticos, hackers o no, eran vistos como científicos locos de pelo salvaje que planeaban la destrucción del mundo o como autómatas de piel pálida y ojos vidriosos, repitiendo frases acartonadas en un tono monótono y aburrido mientras planeaban la siguiente incursión en el gran hermanismo tecnológico.

La mayoría de los hackers optaron por no detenerse en esas impresiones. Pero en 1968 y 1969, los hackers tuvieron que enfrentarse a su triste imagen pública, les gustara o no.

Una marcha de protesta que culminó en Tech Square indicó dramáticamente lo distantes que estaban los hackers de sus pares. Muchos de ellos simpatizaban con la causa contra la guerra. Greenblatt, por ejemplo, había ido a una manifestación en New Haven y había realizado algunas conexiones telefónicas para radicales pacifistas en el Centro Nacional de Información sobre Huelgas en Brandeis. Y el hacker Brian Harvey fue muy activo en la organización de manifestaciones; volvía y contaba en qué baja estima tenían los manifestantes al laboratorio de inteligencia artificial.

Incluso se habló en las reuniones contra la guerra de que algunos de los ordenadores de Tech Square se utilizaban para ayudar a dirigir la guerra. Harvey intentaba decirles que no era así, pero los radicales no sólo no le creían, sino que se enojaban porque intentaba alimentarlos con mentiras.

Los hackers sacudían la cabeza cuando se enteraban de ese desafortunado malentendido. ¡Otro ejemplo de cómo la gente no entendía! Pero una acusación lanzada contra el laboratorio de inteligencia artificial por el movimiento contra la guerra era completamente cierta: todas las actividades del laboratorio, incluso las manifestaciones más estrafalarias o anarquistas de la ética hacker, habían sido financiadas por el Departamento de Defensa. Todo, desde el Sistema Incompatible de Tiempo Compartido hasta el hackeo del metro de Peter Samson, fue pagado por el mismo Departamento de Defensa que estaba matando vietnamitas y reclutando a niños estadounidenses para morir en el extranjero.

La respuesta general del laboratorio de IA a esa acusación fue que la Agencia de Proyectos de Investigación Avanzada (ARPA) del Departamento de Defensa, que financió el laboratorio, nunca le pidió a nadie que ideara aplicaciones militares específicas para la investigación informática realizada por hackers y planificadores. ARPA había sido dirigida por científicos informáticos; su objetivo había sido el avance de la investigación pura. A finales de los años 60, un planificador llamado Robert Taylor estaba a cargo de la financiación de la ARPA, y más tarde admitió haber desviado fondos de proyectos militares "orientados a misiones" a proyectos que harían avanzar la ciencia informática pura. Fue sólo el hacker más raro que llamó a la financiación de la ARPA "dinero sucio".

Casi todos los demás, incluso las personas que se oponían a la guerra, reconocieron que el dinero de la ARPA era el alma del estilo de vida de los hackers. Cuando alguien señaló lo obvio -que el Departamento de Defensa podría no haber pedido aplicaciones militares específicas para el trabajo de inteligencia artificial y sistemas que se estaba haciendo, pero aún así esperaba una bonanza de aplicaciones militares que vendrían de ese trabajo (¿quién podía decir que todo ese trabajo "interesante" en visión y robótica no resultaría en bombardeos más eficientes?) - los hackers negarían lo obvio (Greenblatt: "Aunque nuestro dinero venía del Departamento de Defensa, no era militar") o hablarían como Marvin Minsky: "No hay nada ilegal en que un Departamento de Defensa financie una investigación. Sin duda es mejor que un Departamento de Comercio o de Educación que financie la investigación... porque eso llevaría al control del pensamiento. Preferiría mucho más que los militares estuvieran a cargo de eso... los militares no ocultan lo que quieren, así que no estamos bajo ninguna presión sutil. Está claro lo que está pasando. El caso de ARPA fue único, porque sentían que lo que este país necesitaba era gente buena en tecnología de defensa. En caso de que alguna vez la necesitáramos, la tendríamos".

Los planificadores pensaron que estaban avanzando en la ciencia verdadera. Los hackers estaban formulando alegremente su ordenada filosofía de la nueva era basada en el libre flujo de

información, la descentralización y la democracia informática. Pero los manifestantes antimilitaristas pensaron que era una farsa, ya que todo ese supuesto idealismo beneficiaría en última instancia a la máquina de guerra que era el Departamento de Defensa. La gente contra la guerra quería mostrar su descontento, y un día se filtró la noticia al laboratorio de Inteligencia Artificial de que los manifestantes estaban planeando una marcha que terminaría con una manifestación allí mismo en el noveno piso. Allí, los manifestantes se reunían para demostrar vívidamente que todos ellos (hackers, planificadores y usuarios) eran marionetas del Departamento de Defensa.

Russ Noftsker, el administrador del laboratorio de IA, se tomó muy en serio la amenaza de los manifestantes. Eran los días de Weather Underground y temía que algunos radicales desquiciados estuvieran planeando hacer estallar el ordenador. Se sintió obligado a tomar ciertas medidas para proteger el laboratorio.

Algunas de las medidas eran tan secretas (quizás implicaban a agencias gubernamentales como la CIA, que tenía una oficina en Tech Square) que Noftsker no las revelaba, ni siquiera una década después de que terminara la guerra. Pero otras medidas eran incómodamente obvias. Quitó el cristal de las puertas que conducían desde el vestíbulo del ascensor en el noveno piso hasta la zona donde los hackers jugaban con los ordenadores. En lugar del cristal, Noftsker instaló placas de acero, cubriéndolas con madera para que no pareciera que la zona estaba tan barricada como en realidad lo estaba. Los paneles de vidrio junto a la puerta fueron reemplazados por plexiglás a prueba de balas de media pulgada de espesor para que se pudiera ver quién solicitaba la entrada antes de desbloquear las cerraduras y quitar los pestillos. Noftsker también se aseguró de que las puertas tuvieran bisagras de alta resistencia atornilladas a las paredes, para que los manifestantes no intentaran quitar toda la puerta, entrar corriendo y asaltar las computadoras.

Durante los días previos a la manifestación, solo las personas cuyos nombres estaban en una lista aprobada tenían permitido oficialmente la entrada a esta fortaleza cerrada. El día de la manifestación, incluso llegó al punto de distribuir alrededor de cuarenta cámaras Instamatic a varias personas, pidiéndoles que tomaran fotografías de los manifestantes cuando se aventuraran fuera del área protegida. Si los manifestantes decidían volverse violentos, al menos habría documentación de los malhechores.

Las barricadas funcionaron en la medida en que los manifestantes (alrededor de veinte o treinta de ellos, según los cálculos de Noftsker) caminaron hasta Tech Square, se quedaron afuera del laboratorio un rato y se fueron sin nivelar el PDP-6 con mazos. Pero el suspiro colectivo de alivio por parte de los hackers debe haber estado mezclado con mucho pesar. Si bien habían creado un sistema democrático sin cerraduras dentro del laboratorio, los hackers estaban tan alienados del mundo exterior que tuvieron que usar esas mismas odiadas cerraduras, barricadas y listas compiladas por burócratas para controlar el acceso a este entorno idealista. Si bien algunos podrían haberse quejado por la presencia de las cerraduras, el fervor guerrillero habitual del libre acceso no pareció aplicarse en este caso. Algunos de los hackers, conmocionados por la posibilidad de una derrota, incluso manipularon el sistema de ascensores para que los ascensores no pudieran ir directamente al noveno piso. Aunque anteriormente algunos de los hackers habían declarado: "No trabajaré en un lugar que tenga cerraduras", después de que terminaron las manifestaciones y después de las listas de restricción hubiesen desaparecido hacía tiempo, pero los candados seguían ahí. En general, los hackers optaban por no ver los candados como símbolos de lo alejados que estaban de la corriente dominante.

En el noveno piso reinaba un solipsismo muy decidido, un solipsismo que se mantenía firme incluso cuando el hackerismo sufría algunos ataques directos, aunque ciertamente menos amenazantes físicamente, en publicaciones y revistas. Sin embargo, era difícil ignorar el más cruel de ellos, ya que provenía del interior del MIT, de un profesor de Ciencias de la Computación (sí, el MIT había cambiado de opinión y había creado un departamento) llamado Joseph Weizenbaum. Weizenbaum, un ex programador, un hombre delgado y bigotudo que hablaba con un marcado acento de Europa del Este, había estado en el MIT desde 1963, pero rara vez había interactuado con los hackers. Su mayor contribución en programación a la IA había sido un programa llamado ELIZA, que mantenía una conversación con el usuario; la computadora asumía el papel de un terapeuta. Weizenbaum reconoció el poder de la computadora y se sintió perturbado al observar la seriedad con la que los usuarios interactuaban con ELIZA. Aunque la gente sabía que era "solo" un programa de computadora, le contaban sus secretos más personales. Para Weizenbaum, era una demostración de cómo el poder de la computadora podía llevar a un comportamiento irracional, casi adictivo, con consecuencias deshumanizantes. Y Weizenbaum pensaba que los hackers –o "programadores compulsivos"– eran el colmo de la deshumanización informática. En lo que se convertiría en un pasaje notorio, escribió, en Computer Power and Human Reason:

> . . . se puede ver a jóvenes brillantes de aspecto desaliñado, a menudo con los ojos hundidos y brillantes, sentados ante consolas de computadora, con los brazos tensos y esperando disparar sus dedos, ya preparados para golpear, a los botones y teclas en los que su atención parece estar fijada como la de un jugador en los dados que ruedan. Cuando no están tan absortos, suelen sentarse en mesas llenas de impresiones de ordenador que estudian como estudiantes poseídos de un texto cabalístico. Trabajan hasta casi caerse, veinte o treinta horas seguidas. Si ellos mismos se las arreglan, les traen la comida: café, Coca-Cola, bocadillos. Si es posible, duermen en catres cerca de las impresiones. Sus ropas arrugadas, sus caras sin lavar ni afeitar y su pelo despeinado dan testimonio de que son ajenos a sus cuerpos y al mundo en el que se mueven. Son vagabundos informáticos, programadores compulsivos...

Weizenbaum diría más tarde que la viveza de esta descripción procedía de su propia experiencia como una especie de hacker, y no se basaba directamente en observaciones de la cultura del noveno piso. Pero muchos hackers pensaban lo contrario. Varios pensaron que Weizenbaum los había identificado personalmente, incluso invadiendo su privacidad en su descripción. Otros supusieron que Greenblatt había sido señalado injustamente; De hecho, Greenblatt envió a Weizenbaum algunos mensajes en los que se oponía a la diatriba.

Aun así, no se produjo una introspección general a raíz de este o cualquier otro ataque al estilo de vida de los hackers. Ese no era el modo de ser del laboratorio. Los hackers no solían ahondar en la composición psicológica de los demás. "Había un conjunto de objetivos compartidos", explicaría más tarde Tom Knight, "un conjunto de entusiasmo intelectual compartido, incluso en gran medida un conjunto de vida social compartida, pero también había un límite que la gente no quería traspasar".

Fue este límite tácito el que llegó a molestar al hacker David Silver. Se unió al laboratorio cuando era adolescente y allí literalmente alcanzó la madurez, y además de su productivo hackeo, pasó tiempo pensando en la relación entre los hackers y las computadoras. Llegó a estar fascinado por cómo todos ellos se apegaban tanto, se conectaban tan íntimamente con algo tan simple como el PDP-6. Era casi aterrador: pensar en esto hizo que David Silver se preguntara qué era lo que conectaba a las personas, cómo se encontraban entre sí, por qué se llevaban bien... Cuando algo

relativamente simple como el PDP-6 atrajo tanto a los hackers, todo el tema le hizo preguntarse, por un lado, si las personas eran simplemente computadoras sofisticadas o, por otro, si eran imágenes de Dios como espíritu.

Estas introspecciones no eran cosas que necesariamente compartiera con sus mentores, como Greenblatt o Gosper. “No creo que la gente tuviera conversaciones cálidas entre sí”, diría más tarde. “Ése no era el enfoque. El enfoque estaba en la capacidad intelectual pura”. Este fue el caso incluso con Gosper: el aprendizaje de Silver con él no fue tanto una relación humana cálida, reflexionaría más tarde, como “una relación de hacker”, muy cercana en términos de lo que compartían en términos de la computadora, pero no imbuida de la riqueza de una amistad del mundo real.

“Pasaron muchos, muchos, muchos años en los que todo lo que hice fue hackear computadoras, y no me sentí solo, como si me faltara algo”, diría Silver. “Pero supongo que a medida que empecé a crecer más, a desarrollarme más, a cambiar más, a volverme menos excéntrico en ciertos aspectos, comencé a necesitar más aportes de la gente. [Al no ir a la escuela secundaria] “Dejé de lado todo eso de las relaciones sociales y me metí de lleno en un laboratorio de ideas de cielo azul… Me pasé la vida caminando por ahí hablando como un robot, hablando con un montón de otros robots”.

A veces, el fracaso de los hackers a la hora de ser profundamente personales tenía consecuencias nefastas. El laboratorio podía haber sido el lugar ideal para hackers de nivel gurú, pero para algunos la presión era demasiada. Incluso la disposición física del lugar fomentaba una cierta sensación de alta tensión, con los terminales abiertos, la presencia constante e intimidante de los mejores programadores informáticos del mundo, el aire frío y el zumbido interminable de los aparatos de aire acondicionado. En un momento dado, se llamó a una empresa de investigación para que hiciera un estudio del ruido excesivo e inevitable, y llegaron a la conclusión de que el zumbido del aire acondicionado era tan molesto porque no había suficientes ruidos en competencia, así que arreglaron las máquinas para que emitieran un siseo fuerte y continuo. En palabras de Greenblatt, este cambio “no fue una victoria”, y el siseo constante hizo que las largas horas en el noveno piso fueran bastante estresantes para algunos. Si a eso añadimos otros factores (falta de sueño, comidas salteadas hasta el punto de la desnutrición y una pasión imperiosa por terminar el hack) queda claro por qué algunos hackers se pasan de la raya.

Greenblatt era el que mejor detectaba “el síndrome clásico de varios tipos de pérdidas”, como él lo llamaba. “En cierto modo, me preocupaba el hecho de que no podíamos tener gente cayendo muerta por todas partes”. A veces Greenblatt decía a la gente que se fuera a casa un rato, que se lo tomara con calma. Había otras cosas que estaban fuera de su alcance. Por ejemplo, las drogas. Una noche, mientras conducía de regreso de una comida china, un joven hacker se volvió hacia él y le preguntó, sin bromear, si quería “inyectarse”. Greenblatt se quedó estupefacto. El mundo real estaba penetrando de nuevo y poco podía hacer. Una noche, poco después, ese hacker en particular saltó del puente de Harvard al río Charles cubierto de hielo y resultó gravemente herido. No fue el único intento de suicidio de un hacker de un laboratorio de inteligencia artificial.

A partir de esa sola evidencia, parecería que el argumento de Weizenbaum era válido. Pero había mucho más que eso. Weizenbaum no reconocía la belleza de la devoción hacker en sí misma... ni el idealismo mismo de la ética hacker. No había visto, como Ed Fredkin, a Stew Nelson escribiendo código en el editor TECO mientras Greenblatt y Gosper observaban: sin que ninguno de los tres dijera una palabra, Nelson estaba entreteniendo a los demás, codificando trucos en lenguaje ensamblador que para ellos, con su absoluto dominio de ese "lenguaje" PDP-6, tenían el mismo

efecto que chistes hilarantemente incisivos. Y después de cada pocas instrucciones, había otro chiste en esta sublime forma de comunicación... La escena era una demostración de compartir que Fredkin nunca olvidó.

Si bien admitió que las relaciones entre hackers eran inusuales, especialmente porque la mayoría de los hackers vivían vidas asexuales, Fredkin diría más tarde que "estaban viviendo el futuro de las computadoras... Simplemente se divertían". Sabían que eran de élite, algo especial. Y creo que se apreciaban mutuamente. Todos eran diferentes, pero cada uno sabía algo bueno del otro. Todos se respetaban. No sé si algo parecido [a esa cultura hacker] ha sucedido en el mundo. Yo diría que se amaban entre sí".

Los hackers se centraban en la magia de las computadoras en lugar de las emociones humanas, pero también podían ser tocados por otras personas. Un buen ejemplo sería el caso de Louis Merton (un seudónimo). Merton era un estudiante del MIT, algo reservado y un jugador de ajedrez excepcional. Salvo por este último rasgo, Greenblatt al principio pensó que estaba dentro del espectro de personas al azar que podrían deambular por el laboratorio.

El hecho de que Merton fuera un jugador de ajedrez tan bueno agradó a Greenblatt, que entonces estaba trabajando para construir una computadora real que ejecutara una versión mejorada de su programa de ajedrez. Merton aprendió algo de programación y se unió a Greenblatt en el proyecto. Más tarde hizo su propio programa de ajedrez en un PDP-7 poco utilizado en el noveno piso. Merton era un entusiasta del ajedrez y de las computadoras, y no había nada que presagiara lo que sucedió durante las vacaciones de Acción de Gracias a fines de 1966, cuando, en la pequeña sala de juegos de IA con apariencia de teatro en el octavo piso de Tech Square (donde el profesor Seymour Papert y un grupo estaban trabajando en el lenguaje informático educativo LOGO), Merton se convirtió temporalmente en un vegetal. Asumió una posición clásica de catatonia, sentado rígidamente erguido, con las manos apretadas en puños a los costados. No respondía a las preguntas, ni siquiera reconocía la existencia de nada fuera de él. La gente no sabía qué hacer. Llamaron a la enfermería del MIT y les dijeron que llamaran a la policía de Cambridge, que se llevó al pobre Merton. El incidente sacudió severamente a los hackers, incluido Greenblatt, quien se enteró cuando regresó de una visita de vacaciones a su casa.

Merton no era uno de los principales hackers. Greenblatt no era un amigo íntimo. No obstante, Greenblatt se dirigió inmediatamente a la Hospital Estatal de Westboro para buscar a Merton. El viaje fue largo y el destino le recordó a Greenblatt algo de la Edad Media: más que un hospital, una prisión. Greenblatt decidió no irse hasta que Merton saliera. El último paso de este tortuoso proceso fue conseguir la firma de un médico anciano, aparentemente senil. “Exactamente como algo sacado de una película de terror”, recordó Greenblatt más tarde. “No sabía leer. Este tipo de asistente al azar le decía: ‘Firme aquí. Firme aquí’”.

Resultó que Merton tenía antecedentes de estos problemas. A diferencia de la mayoría de los catatónicos, Merton mejoraba después de unos días, especialmente cuando le daban medicamentos. A menudo, cuando se volvía catatónico en algún lugar, quien lo encontraba llamaba a alguien para que se lo llevara, y los médicos daban un diagnóstico de catatonia permanente incluso cuando Merton estaba volviendo a la vida. Él llamaba al laboratorio de inteligencia artificial y decía: “Ayuda”, y alguien, a menudo Greenblatt, venía a buscarlo. Más tarde, alguien descubrió en los registros del MIT una carta de la difunta madre de Merton. La carta explicaba que Louis era un chico extraño y que a veces se ponía rígido. En ese caso, todo lo que había que hacer era preguntar:

«Louis, ¿te gustaría jugar una partida de ajedrez?». Fredkin, que también se había interesado mucho por Merton, lo intentó. Un día Merton se puso rígido en el borde de su silla, totalmente en modo escultural. Fredkin le preguntó si le gustaría jugar al ajedrez, y Merton marchó rígidamente hacia el tablero de ajedrez. La partida comenzó con Fredkin charlando en una conversación bastante unilateral, pero de repente Merton se detuvo. Fredkin preguntó: «Louis, ¿por qué no mueves?». Después de una pausa muy larga, Merton respondió con una voz gutural y lenta: «Tu... rey está... en... jaque». Fredkin había descubierto inadvertidamente el jaque de su último movimiento.

La condición de Merton podía ser mitigada con cierta medicina, pero por razones propias casi nunca la tomaba. Greenblatt le suplicaba, pero él se negaba. Una vez Greenblatt fue a ver a Fredkin para pedirle ayuda; Fredkin regresó con Greenblatt y encontró a Merton rígido e insensible.

“Louis, ¿cómo es que no estás tomando tu medicina?”, preguntó. Merton se quedó sentado allí, con una débil sonrisa congelada en su rostro. “¿Por qué no la tomas?”, repitió Fredkin.

De repente, Merton se echó hacia atrás y golpeó a Fredkin en la barbilla. Ese tipo de comportamiento era una de las características desafortunadas de Merton. Pero los hackers mostraron una tolerancia notable. No lo descartaron como un perdedor. Fredkin consideró el caso de Merton un buen ejemplo de la humanidad esencial del grupo que Weizenbaum había, en efecto, descartado como androides sin emociones. “Está loco”, diría Minsky más tarde sobre Weizenbaum. “Estos [los hackers] son las personas más sensibles y honorables que han existido jamás”. Tal vez sea una hipérbole, pero era cierto que detrás de su determinación había calidez, en la realización colectiva de la Ética Hacker. Como cualquier orden religiosa devota, los hackers habían sacrificado lo que los de afuera considerarían un comportamiento emocional básico, por amor al hacking.

David Silver, quien eventualmente abandonaría la orden, todavía estaba asombrado por ese hermoso sacrificio años después: “Era en cierto modo necesario que estas personas fueran extremadamente brillantes y, en cierto sentido, socialmente discapacitadas para que se concentraran en una sola cosa”. El hacking. Lo más importante del mundo para ellos.

• • • • • • •

El mundo de la informática fuera de Cambridge no se quedó quieto mientras la Ética Hacker florecía en el noveno piso de Tech Square. A finales de los años 60, el hackerismo se estaba extendiendo, en parte debido a la proliferación de máquinas interactivas como la PDP-10 o la XDS-940, en parte debido a los entornos de programación amigables (como el que los hackers habían creado en el MIT) y en parte porque los veteranos del MIT dejaban el laboratorio y llevaban su cultura a nuevos lugares. Pero el núcleo del movimiento era esto: la gente que quería hackear estaba encontrando computadoras en las que hackear.

Estas computadoras no estaban necesariamente en el MIT. Los centros de cultura hacker estaban creciendo en varias instituciones en todo el país, desde Stanford hasta Carnegie-Mellon. Y a medida que estos otros centros alcanzaron una masa crítica (suficiente gente dedicada para hackear un gran sistema y hacer peregrinaciones nocturnas a restaurantes chinos locales), se volvieron lo suficientemente tentadores como para alejar a algunos de los hackers del laboratorio de IA de Tech Square. El intenso estilo de hackerismo del MIT se exportaría a través de estos emisarios.

A veces, los hackers no se mudaban a una institución, sino a una empresa. Un programador llamado Mike Levitt fundó una empresa de tecnología de vanguardia llamada Systems Concepts en San Francisco. Fue lo suficientemente inteligente como para contratar como socio a Stew Nelson, un

hacker especializado en teléfonos y PDP-1; el maestro de la música TX-0 Peter Samson también se unió a esta empresa de diseño y fabricación de hardware de alta tecnología. En conjunto, la pequeña empresa logró trasladar a San Francisco a gran parte del talento concentrado en Tech Square. No fue una hazaña fácil, ya que los hackers generalmente se oponían a la exigencia de la vida en California, particularmente conducir, y exposición recreativa al sol. Pero Nelso había aprendido antes su lección, a pesar de las repetidas insistencias de Fredkin a mediados de los sesenta, se había negado a ir a la nueva sede de Triple-I en Los Ángeles hasta que un día, después de reiterar enfáticamente su promesa, salió furioso de Tech Square sin abrigo. Era el día más frío del invierno de Cambridge ese año, y tan pronto como salió a la calle, sus gafas se rompieron por el cambio repentino de temperatura. Caminó directamente de regreso a la oficina de Fredkin, con las cejas cubiertas de carámbanos, y dijo: "Me voy a Los Ángeles".

En algunos casos, la partida de un hacker se aceleraba por lo que Minsky y Ed Fredkin llamaban "ingeniería social". A veces, los planificadores encontraban que un hacker se estaba estancando, tal vez atascado en algún problema de sistemas, o tal vez se obsesionaba tanto con actividades extracurriculares, como hackear cerrojos o teléfonos, que los planificadores consideraban que su trabajo ya no era "interesante". Fredkin recordaría más tarde que los hackers podían llegar a un estado en el que eran “como anclas que arrastraban la cosa hacia abajo. El tiempo había pasado, en cierto sentido. Necesitaban salir del laboratorio y el laboratorio los necesitaba fuera. Así que alguna oferta sorprendente llegaba a esas personas, o alguna visita organizada, generalmente en algún lugar muy, muy lejano. Esas personas comenzaban a filtrarse en el mundo hacia empresas u otros laboratorios. No era el destino, yo lo organizaba”.

Minsky diría, “Valiente Fredkin”, reconociendo la naturaleza clandestina de la actividad de Fredkin, que tendría que realizarse sin el conocimiento de la comunidad hacker; no tolerarían una estructura organizativa que realmente dictara a dónde debía ir la gente.

Si bien el destino podía ser la industria (además de Systems Concepts, la empresa Information International de Fredkin contrató a muchos de los hackers del MIT), a menudo era otro centro informático. El más deseable de ellos era el Laboratorio de Inteligencia Artificial de Stanford (SAIL), que el tío John McCarthy había fundado cuando dejó el MIT en 1962.

En muchos aspectos, SAIL era una imagen especular del funcionamiento del MIT, distorsionada sólo por la neblina californiana que a veces se desplazaba desde el océano Pacífico hasta la península. Pero la distorsión de California era significativa, demostrando que incluso lo más parecido a la comunidad hacker del MIT era sólo una aproximación del ideal; el estilo de invernadero del hackerismo del MIT estaba destinado a viajar, pero cuando se exponía a cosas como la luz del sol de California, perdía un poco de intensidad.

La diferencia empezaba con el entorno, un antiguo centro de conferencias semicircular de hormigón, vidrio y madera de secuoya en las colinas que dominaban el campus de Stanford. Dentro del edificio, los hackers trabajarían en cualquiera de las sesenta y cuatro terminales repartidas por las distintas oficinas. Nada de la claustrofobia de Tech Square. Sin ascensores, sin el ensordecedor siseo del aire acondicionado. El estilo relajado hizo que gran parte de la acritud a veces constructiva del MIT (las sesiones de gritos en el aula del TMRC, las guerras religiosas entre estudiantes de posgrado y hackers) no se trasladara. En lugar de las imágenes llenas de batallas de la ciencia ficción espacial de disparos que impregnaban Tech Square, las imágenes de Stanford eran la amable tradición de elfos, hobbits y magos descritos en la trilogía de la Tierra Media de J.R.R. Tolkien. Las

salas del laboratorio de inteligencia artificial llevaban nombres de lugares de la Tierra Media, y la impresora SAIL estaba configurada para que pudiera manejar tres fuentes élficas diferentes.

La diferencia de California se reflejó en el famoso género de juegos de computadora que el laboratorio de Stanford finalmente desarrolló después del apogeo de Spacewar del MIT. Un hacker de Stanford llamado Donald Woods descubrió un día una especie de juego en una computadora de investigación Xerox que involucraba a un espeleólogo-explorador que buscaba un tesoro en una mazmorra. Woods se puso en contacto con el programador, Will Crowther, habló con él sobre el tema y decidió ampliar el juego de Crowther para convertirlo en una aventura a gran escala, en la que una persona podría usar la computadora para asumir el papel de un viajero en un entorno al estilo de Tolkien, luchar contra enemigos, superar obstáculos mediante trucos ingeniosos y, finalmente, recuperar un tesoro. El jugador daría órdenes de dos palabras, verbo-sustantivo, al programa, que respondería dependiendo de cómo la orden cambiara el universo que había sido creado dentro de la computadora por la imaginación de Don Woods. Por ejemplo, el juego comenzaba con la computadora describiendo tu ubicación inicial:

> ESTÁS PARADO AL FINAL DE UN CAMINO ANTE UN PEQUEÑO EDIFICIO DE LADRILLOS. A TU ALREDEDOR HAY UN BOSQUE. UN PEQUEÑO ARROYO FLUYE FUERA DEL EDIFICIO Y BAJA POR UN BARRANCO.

Si escribías IR AL SUR, la computadora diría:

> ESTÁS EN UN VALLE EN EL BOSQUE AL LADO DE UN ARROYO QUE CAE POR UN LECHO ROCOSO.

Más adelante, tendrías que idear todo tipo de trucos para sobrevivir. Por ejemplo, la serpiente con la que te topabas solo podía eliminarse liberando un pájaro que habías recogido por el camino. El pájaro atacaría a la serpiente y tú serías libre de pasar. Cada "sala" de la aventura era como una subrutina de computadora que presentaba un problema lógico que debías resolver.

En cierto sentido, Adventure era una metáfora de la programación informática en sí: los recovecos profundos que explorabas en el mundo de Adventure eran similares a los niveles más básicos y oscuros de la máquina por la que viajabas cuando hackeabas en código ensamblador. Te podías marear tratando de recordar dónde estabas en ambas actividades. De hecho, Adventure resultó tan adictivo como la programación: Woods puso el programa en el SAIL PDP-10 un viernes, y algunos hackers (y "turistas" del mundo real) pasaron todo el fin de semana tratando de resolverlo. Como cualquier buen sistema o programa, por supuesto, Adventure nunca estaba terminado: Woods y sus amigos siempre lo mejoraban, lo depuraban, añadían más acertijos y funciones. Y como todo programa importante, Adventure expresaba la personalidad y el entorno de los autores. Por ejemplo, la visión de Woods de un puente de peaje cubierto de niebla protegido por un troll testarudo llegó durante un descanso en el hackeo una noche, cuando Woods y otros hackers decidieron ver el amanecer en un Monte Diablo envuelto en niebla, a una distancia considerable en coche. No llegaron a tiempo, y Woods recordó cómo era ese amanecer brumoso y lo escribió en la descripción de esa escena del juego, que concibió durante el desayuno esa mañana.

Fue en Stanford donde los gurús tenían la misma probabilidad de ser miembros de la facultad que hackers de sistemas (entre los profesores de Stanford estaba el famoso científico informático Donald Knuth, autor del clásico de varios volúmenes El arte de la programación informática). Fue en Stanford donde, antes de la locura de Adventure, los placeres casuales de Spacewar se

perfeccionaron hasta convertirse en un gran arte (Slug Russell había salido con McCarthy, pero fueron hackers más jóvenes los que desarrollaron versiones para cinco jugadores y opciones para la reencarnación, y organizaron extensos torneos que duraban toda la noche). Fue en Stanford donde los hackers realmente dejaban sus terminales para jugar un partido diario de voleibol. Fue en Stanford donde se llevó a cabo con éxito una campaña de recaudación de fondos para una ampliación del laboratorio, algo que habría sido inconcebible en el MIT: una sauna. Fue en Stanford donde la computadora pudo soportar imágenes de video, lo que permitió a los usuarios cambiar de un programa de computadora a un programa de televisión. El uso más famoso de esto, según algunos habituales de SAIL, se produjo cuando los hackers de SAIL pusieron un anuncio en el periódico del campus buscando una pareja de jóvenes universitarias dispuestas a colaborar. Las mujeres que respondieron al anuncio se convirtieron en estrellas de una orgía sexual en el laboratorio de IA, captadas por una cámara de vídeo y vigiladas por los terminales por hackers agradecidos. Algo que nunca habría ocurrido en el MIT.

No es que los hackers de SAIL fueran menos dedicados a su actividad que la gente del MIT. En un artículo que resume la historia del laboratorio de Stanford, el profesor Bruce Buchanan se refiere al "extraño entorno social creado por jóvenes intensos cuyo primer amor fue el hacking", y era cierto que los extremos a los que llegaban los hackers en California no eran menos que los de Tech Square. Por ejemplo, los hackers de SAIL no tardaron mucho en darse cuenta de que el espacio de arrastre entre el techo artificial bajo y el tejado podía ser una cómoda conejera para dormir, y varios de ellos vivieron allí durante años. Un hacker de sistemas pasó los primeros años de la década de 1970 viviendo en su coche averiado aparcado en el aparcamiento de fuera del edificio; una vez a la semana iba en bicicleta a Palo Alto a por provisiones. La otra alternativa para conseguir comida era el Prancing Pony; llamado así por una taberna de la Tierra Media, era la máquina expendedora de comida de SAIL, llena de delicias de comida sana y empanadillas chinas de un restaurante chino local. Cada hacker tenía una cuenta en el Prancing Pony, que era administrada por el ordenador. Después de hacer la compra de comida, se le daba la opción de duplicar o nada el coste de la comida, y el resultado dependía de si era un milisegundo par o impar cuando hacía la apuesta. Con ese tipo de disposiciones, SAIL era incluso más receptivo que el MIT para el hackeo las 24 horas del día. Tenía su gente de aplicaciones y su gente de sistemas. Estaba abierto a los forasteros, que se sentaban y empezaban a hackear; y si demostraban ser prometedores, el tío John McCarthy podía contratarlos.

Los hackers de SAIL también vivían según la ética hacker. El sistema de tiempo compartido de la máquina SAIL, al igual que el ITS, no requería contraseñas, pero, por insistencia de John McCarthy, el usuario tenía la opción de mantener sus archivos privados. Los hackers de SAIL escribieron un programa para identificar a estas personas y procedieron a desbloquear los archivos, que leerían con especial interés. “Cualquiera que pida privacidad debe estar haciendo algo interesante”, explicaría más tarde el hacker de SAIL Don Woods.

Asimismo, SAIL no era en absoluto inferior al MIT en cuanto a realizar un trabajo informático importante. Al igual que sus homólogos del laboratorio de IA del MIT, los hackers de SAIL eran fanáticos de la robótica, como lo implicaba el cartel del exterior de SAIL: PRECAUCIÓN, VEHÍCULO ROBOT. El sueño de John McCarthy era que un robot saliera del extraño laboratorio de IA y recorriera los cinco kilómetros hasta el campus con su propio poder físico y mental. En un momento dado, presumiblemente por error, un robot se soltó y se dirigía cuesta abajo cuando, por fortuna, en un momento dado, un trabajador que se dirigía al laboratorio lo vio y lo rescató. Varios

hackers y académicos trabajaban en SAIL en importantes campos de planificación como la comprensión del habla y los estudios del lenguaje natural. Algunos de los hackers se involucraron mucho en un proyecto de música por computadora que abriría camino en ese campo.

Stanford y otros laboratorios, ya sea en universidades como Carnegie-Mellon o centros de investigación como Stanford Research Institute, se acercaron entre sí cuando ARPA conectó sus sistemas informáticos a través de una red de comunicaciones. Esta "ARPAnet" estaba muy influenciada por la ética hacker, ya que entre sus valores estaba la creencia de que los sistemas deberían estar descentralizados, alentar la exploración e instar a un flujo libre de información. Desde un ordenador en cualquier "nodo" de ARPAnet, se podía trabajar como si se estuviera sentado en una terminal de un sistema informático distante. Hackers de todo el país podían trabajar en el sistema ITS en Tech Square, y los valores hacker implícitos en eso se estaban extendiendo. La gente se enviaba un enorme volumen de correo electrónico, intercambiaba conocimientos técnicos esotéricos, colaboraba en proyectos, jugaba a la aventura, forjaba amistades cercanas con hackers que no conocía en persona y mantenía contacto con amigos en lugares en los que había hackeado anteriormente. El contacto ayudó a normalizar el hackerismo, de modo que se podía encontrar hackers en Utah hablando en la peculiar jerga desarrollada en la Sala de Herramientas junto al Tech Model Railroad Club.

Sin embargo, incluso cuando la Ética Hacker creció en el número real de sus adeptos, los hackers del MIT notaron que fuera de Cambridge las cosas no eran iguales. El hackerismo de Greenblatt, Gosper y Nelson se había orientado demasiado a la creación de una utopía, e incluso las ramificaciones muy similares estaban, en comparación, perdiendo en varios sentidos.

“¿Cómo pudiste ir a California, lejos de la acción?”, preguntaba la gente a los que iban a Stanford. Algunos se fueron porque se cansaron de la dicotomía ganador-perdedor en el noveno piso, aunque admitían que la intensidad del MIT no estaba en California. Tom Knight, que hackeó en Stanford durante un tiempo, solía decir que en Stanford no se podía hacer un buen trabajo.

David Silver también fue allí y llegó a la conclusión de que “la gente de Stanford era una especie de perdedores en su forma de pensar. No eran tan rigurosos en ciertos aspectos y eran más amantes de la diversión. Uno estaba construyendo un coche de carreras y otro estaba construyendo un avión en el sótano…”. El propio Silver se metió en el mundo del hardware en Stanford cuando construyó un conmutador de audio que permitía a la gente que trabajaba en sus terminales escuchar cualquiera de los dieciséis canales, desde estaciones de radio hasta un sistema de megafonía SAIL. Todas las opciones, por supuesto, se almacenaban en el SAIL PDP-6. Y Silver cree que la exposición al estilo de hackeo de California le ayudó a relajarse, preparándole para romper con la sociedad cerrada del noveno piso.

La deserción de Silver y los demás hackers del MIT no paralizó el laboratorio. Llegaron nuevos hackers para sustituirlos. Greenblatt y Gosper se quedaron, al igual que Knight y algunos otros hackers canónicos. Pero la energía tremendamente optimista que se generó con la explosión inicial de la investigación en IA, con la creación de nuevos sistemas de software, parecía haberse disipado. Algunos científicos se quejaban de que las fanfarronadas de los primeros planificadores de IA no se habían cumplido. Dentro de la propia comunidad hacker, los fervientes hábitos y los extraños patrones establecidos en la última década parecían haberse solidificado. ¿Se habían osificado también? ¿Se podía envejecer como hacker y seguir trabajando en esas jornadas de treinta horas? “Estaba muy orgulloso”, diría Gosper más tarde, “de poder hackear las veinticuatro horas del día y

no preocuparme realmente en qué fase del sol o la luna estaba. Despertarme y encontrarme con el crepúsculo, sin tener idea de si era el amanecer o el atardecer". Sin embargo, sabía que no podía continuar eternamente. Y cuando no pudiera, cuando no hubiera Gosper ni Greenblatt lamentándose durante treinta horas, ¿hasta dónde llegaría el sueño hacker? ¿Habría significado realmente algo la Edad de Oro, que ahora se acerca a su fin?

• • • • • • •

Fue en 1970 cuando Bill Gosper comenzó a hackear LIFE. Era otro sistema que era un mundo en sí mismo, un mundo donde el comportamiento era "extremadamente rico, pero no tan rico como para ser incomprensible". Esto obsesionaría a Bill Gosper durante años.

LIFE era un juego, una simulación por computadora desarrollada por John Conway, un distinguido matemático británico. Fue descrito por primera vez por Martin Gardner, en su columna "Juegos matemáticos" en la edición de octubre de 1970 de Scientific American. El juego consiste en marcadores en un campo similar a un tablero de ajedrez, cada marcador representa una "célula". El patrón de células cambia con cada movimiento en el juego (llamado "generación"), dependiendo de unas pocas reglas simples: las células mueren, nacen o sobreviven a la siguiente generación según la cantidad de células vecinas que haya en las inmediaciones. El principio es que las células aisladas mueren de soledad y las células hacinadas mueren de superpoblación; las condiciones favorables generarán nuevas células y mantendrán vivas a las antiguas. La columna de Gardner habló de las complejidades que se hacían posibles gracias a este sencillo juego y postuló algunos resultados extraños que Conway o sus colaboradores aún no habían logrado.

Gosper vio el juego por primera vez cuando entró en el laboratorio un día y encontró a dos hackers jugando con él en el PDP-6. Observó durante un rato. Su primera reacción fue descartar el ejercicio por no ser interesante. Luego observó cómo los patrones tomaban forma durante un rato más. Gosper siempre había apreciado cómo el ancho de banda específico del globo ocular humano podía interpretar patrones; a menudo utilizaba algoritmos extraños para generar una visualización basada en cálculos matemáticos. Lo que parecerían números aleatorios en el papel podían cobrar vida en una pantalla de ordenador. Se podía discernir un cierto orden, un orden que cambiaría de una manera interesante si se llevaba el algoritmo unas cuantas iteraciones más allá o se alternaban los patrones x e y. Gosper pronto tuvo claro que LIFE ofrecía estas posibilidades y más. Comenzó a trabajar con algunos trabajadores de inteligencia artificial para hackear LIFE de una manera extremadamente seria. No iba a hacer casi nada más durante los siguientes dieciocho meses.

El primer esfuerzo del grupo fue tratar de encontrar una configuración en el universo LIFE, que era posible en teoría pero que no había sido descubierta. Por lo general, sin importar con qué patrón se comenzara, después de unas pocas generaciones se agotaría hasta desaparecer o volvería a uno de varios patrones estándar que reciben su nombre de la forma que formaba el conjunto de células. Los patrones incluían la colmena, la granja de miel (cuatro colmenas), la nave espacial, el polvorín, el faro, la cruz latina, el sapo, el molinete y la esvástica. A veces, después de varias generaciones, los patrones se alternaban, parpadeando entre uno y otro: se los llamaba osciladores, semáforos o púlsares. Lo que Gosper y los hackers buscaban se llamaba arma planeadora. Un planeador era un patrón que se movía por la pantalla, volviendo periódicamente a la misma forma puntiaguda. Si alguna vez crearas un patrón LIFE que realmente arrojara planeadores a medida que cambiaba de forma, tendrías un cañón planeador, y el inventor de LIFE, John Conway, ofreció cincuenta dólares a la primera persona que fuera capaz de crear uno.

Los hackers pasarían toda la noche sentados ante la pantalla “340” de alta calidad del PDP-6 (un monitor especial de alta velocidad fabricado por DEC), probando diferentes patrones para ver qué obtenían. Registrarían cada “descubrimiento” que hicieran en este universo artificial en un gran cuaderno de bocetos negro, que Gosper bautizó como el álbum de recortes de LIFE. Mirarían la pantalla mientras, generación tras generación, el patrón iba cambiando. A veces parecía un gusano que chasqueaba la cola entre reveses repentinos, como si estuviera alternando entre sí mismo y un reflejo en un espejo. Otras veces, la pantalla acabaría oscureciéndose a medida que las células morían por sobrepoblación agregada y luego por aislamiento. Un patrón podría terminar con la pantalla en blanco. Otras veces las cosas se detendrían con un patrón estable de “naturaleza muerta” de uno de los estándares. O bien, parecía que las cosas se estaban acabando y una pequeña célula expulsada por una “colonia” moribunda podía alcanzar otro patrón y este recién llegado podía hacerlo explotar de actividad. “Las cosas podían salir corriendo y hacer algo increíblemente aleatorio”, recordaría Gosper más tarde sobre aquellas fantásticas primeras semanas, “y no podíamos dejar de mirarlo. Nos quedábamos sentados allí, preguntándonos si iba a continuar para siempre”.

Mientras jugaban, el mundo que los rodeaba parecía estar conectado en patrones de una simulación de LIFE. A menudo escribían un patrón arbitrario, como el tejido de una prenda de vestir, o un patrón que uno de ellos discernía en una imagen o un libro. Por lo general, lo que hacía no era interesante. Pero a veces detectaban un comportamiento inusual en una pequeña parte de un gran patrón de LIFE. En ese caso, intentaban aislar esa parte, como hacían cuando notaban un patrón que se llamaría “la lanzadera”, que se movía una distancia en la pantalla y luego daba marcha atrás. La lanzadera dejaba atrás algunas células en su camino, que los hackers llamaban “dribbles”. Los regueros eran “veneno” porque su presencia causaría estragos en poblaciones de VIDA que de otro modo serían estables.

Gosper se preguntó qué podría pasar si dos lanzaderas rebotaran una contra la otra, y calculó que había entre doscientas y trescientas posibilidades. Probó cada una de ellas, y finalmente encontró un patrón que realmente hacía que los planeadores se desprendieran. Se movía por la pantalla como un látigo que se movía en un jitterbug, arrojando bumeranes flácidos de fósforo. Era una vista maravillosa. No era de extrañar que esto se llamara VIDA: el programa creaba la vida misma. Para Gosper, la simulación de Conway era una forma de creación genética, sin las viles secreciones y las complicaciones emocionales asociadas con la versión del mundo real de crear nueva vida. ¡Felicitaciones, has dado a luz a un cañón planeador!

Temprano a la mañana siguiente, Gosper se aseguró de imprimir las coordenadas del patrón que resultó en el cañón planeador, y corrió a la oficina de Western Union para enviar un telegrama a Martin Gardner con la noticia. Los hackers se quedaron con los cincuenta dólares.

Esto no acabó de ninguna manera con la locura de LIFE en el noveno piso. Cada noche, Gosper y sus amigos monopolizaban la pantalla 340 que mostraba varios patrones de LIFE, un entretenimiento, exploración y viaje continuos a una existencia alternativa. Algunos no compartían su fascinación, en particular Greenblatt. A principios de los setenta, Greenblatt había asumido un papel más de liderazgo en el laboratorio. Parecía preocuparse más por las cosas que había que hacer y, después de ser el cuidador de facto del sistema ITS, estaba tratando activamente de transformar su visión del sueño hacker en una máquina que lo encarnara. Había dado los primeros pasos en su "máquina de ajedrez", que respondía con una rapidez inaudita en la mayoría de las computadoras.

También estaba tratando de asegurarse de que el laboratorio en sí funcionara sin problemas para que el hackeo progresara y fuera continuamente interesante.

LIFE no le encantaba. En concreto, no estaba contento con que Gosper y los demás pasaran «cantidades increíbles de horas frente a la consola, mirando fijamente esas cosas de LIFE» y monopolizando la única terminal 340. Lo peor de todo era que consideraba que el programa que estaban usando «claramente no era óptimo». Esto era algo que los hackers de LIFE admitieron de buena gana, pero el caso LIFE era el raro ejemplo de hackers que toleraban cierta ineficiencia. Estaban tan emocionados con el despliegue de LIFE que no querían detenerse ni siquiera durante los pocos días que podrían llevar hackear un programa mejor. Greenblatt aulló en protesta —«el nivel de calor llegó a ser moderadamente alto», admitió más tarde— y no se calló hasta que uno de los hackers de LIFE escribió un programa más rápido, cargado de utilidades que le permitían ir hacia atrás y hacia adelante durante un número específico de generaciones, centrarse en varias partes de la pantalla y hacer todo tipo de cosas para mejorar la exploración.

Greenblatt nunca entendió la idea. Pero para Gosper, LIFE era mucho más que un hack normal. Lo veía como una forma de “básicamente hacer ciencia en un nuevo universo en el que todos los tipos inteligentes no te han eliminado hace doscientos o trescientos años. Es la historia de tu vida si eres matemático: cada vez que descubres algo interesante, descubres que Gauss o Newton lo sabían en su cuna. Con LIFE eres el primero en llegar, y siempre están sucediendo cosas divertidas. Puedes hacer de todo, desde la teoría de funciones recursivas hasta la cría de animales. Hay una comunidad de personas que comparten estas experiencias contigo. Y existe la sensación de conexión entre tú y el medio ambiente. La idea de dónde está el límite de un ordenador. ¿Dónde termina el ordenador y dónde empieza el medio ambiente?”.

Obviamente, Gosper estaba hackeando LIFE con una intensidad casi religiosa. Las metáforas implícitas en la simulación (de poblaciones, generaciones, nacimiento, muerte, supervivencia) se estaban volviendo reales para él. Comenzó a preguntarse cuáles serían las consecuencias si un superordenador gigante se dedicara a LIFE… e imaginó que, con el tiempo, podrían crearse algunos objetos improbables a partir del patrón. Los más persistentes entre ellos sobrevivirían contra probabilidades que Gosper, como matemático, sabía que eran casi imposibles. No sería la aleatoriedad la que determinaría la supervivencia, sino una especie de darwinismo informático. En este juego que es una lucha contra la descomposición y el olvido, los sobrevivientes serían los "estados de materia de máxima persistencia". Gosper pensó que estas formas de VIDA se las habrían ingeniado para existir; en realidad, habrían evolucionado hasta convertirse en entidades inteligentes.

"Así como las rocas se desgastan en unos pocos miles de millones de años, pero el ADN permanece allí", explicaría más tarde. "Este comportamiento inteligente sería simplemente otro de esos fenómenos organizativos como el ADN que se las ingeniaron para aumentar la probabilidad de supervivencia de alguna entidad. Por lo tanto, uno tiende a sospechar, si no es creacionista, que configuraciones de VIDA muy, muy grandes eventualmente exhibirían [características] inteligentes. Especular sobre lo que estas cosas podrían saber o podrían descubrir es muy intrigante... y tal vez tenga implicaciones para nuestra propia existencia".

Gosper se sintió estimulado aún más por la teoría de Ed Fredkin de que es imposible saber si el universo no es una simulación por computadora, tal vez dirigida por algún hacker en otra dimensión. Gosper llegó a especular que en su imaginaria máquina de VIDA definitiva, las

entidades inteligentes que se formarían a lo largo de miles de millones de generaciones también podrían participar en esas mismas especulaciones. Según la forma en que entendemos nuestra propia física, es imposible hacer una computadora perfectamente confiable. Entonces, cuando ocurrió un error inevitable en esa supermáquina de VIDA, las entidades inteligentes en la simulación se habrían encontrado de repente con una ventana a la metafísica que determinaba su propia existencia. Tendrían una pista de cómo se implementaron realmente. En ese caso, conjeturó Fredkin, las entidades podrían concluir con precisión que eran parte de una simulación gigante y podrían querer rezar a sus implementadores ordenándose en patrones reconocibles, pidiendo en lenguaje legible que les dieran pistas sobre cómo eran. Gosper recuerda que “eso lo ofendió, que no pude entenderlo en absoluto durante días, antes de aceptarlo”.

Lo aceptó.

Tal vez no sea tan sorprendente. En cierto sentido, esa conjetura lejana ya era una realidad. ¿Qué eran los hackers sino dioses de la información, que movían fragmentos de conocimiento en patrones cósmicamente complejos dentro del PDP-6? ¿Qué los satisfacía más que este poder? Si uno admite que el poder corrompe, entonces uno podría identificar corrupción en el fracaso de los hackers para distribuir este poder (y el sueño hacker en sí) más allá de los límites del laboratorio. Ese poder estaba reservado para los ganadores, un círculo interno que podría vivir según la Ética Hacker pero que hacía poco esfuerzo por ampliar el círculo más allá de los que eran como ellos, impulsados por la curiosidad, el genio y el Imperativo de la Manos a la Obra.

No mucho después de su inmersión en LIFE, el propio Gosper vislumbró los límites del estrecho círculo que habían dibujado los hackers. Sucedió durante la luz artificial del día del lanzamiento a la Luna del Apolo 17 en 1972. Era un pasajero de un crucero especial al Caribe, un “crucero científico” programado para el lanzamiento, y el barco estaba lleno de escritores de ciencia ficción, futuristas, científicos de diversos tipos, comentaristas culturales y, según Gosper, “una increíble cantidad de cruceros completamente tontos”.

Gosper estaba allí como parte del grupo de Marvin Minsky. Tuvo la oportunidad de participar en discusiones con personas como Norman Mailer, Katherine Anne Porter, Isaac Asimov y Carl Sagan, quien impresionó a Gosper con su forma de jugar al ping-pong. Para competir de verdad, Gosper se coló en algunas partidas prohibidas con los tripulantes indonesios, que eran, con diferencia, los mejores jugadores del barco.

El Apolo 17 iba a ser el primer lanzamiento espacial tripulado que se iniciara de noche, y el crucero estaba a tres millas de Cabo Kennedy para disfrutar de una vista ventajosa del lanzamiento. Gosper había oído todos los argumentos en contra de tomarse la molestia de ver un despegue: ¿por qué no verlo por televisión, ya que estaría a kilómetros de la plataforma de lanzamiento real? Pero cuando vio que la maldita cosa despegaba, apreció la distancia. La noche había ardido y el pico de energía le llegó hasta las entrañas. La camisa le golpeó el pecho, el cambio en su bolsillo tintineó y los altavoces del sistema de megafonía se soltaron de sus soportes en el soporte de observación y quedaron colgando de sus cables de alimentación. El cohete, que por supuesto nunca habría podido mantener un rumbo tan exacto sin computadoras, saltó al cielo, empeñado en alcanzar el cosmos como un vengador en llamas, una pesadilla de Spacewar; los pilotos de cruceros quedaron atónitos y en trance ante el poder y la gloria de la vista. Los tripulantes indonesios se volvieron locos. Gosper más tarde los recordó corriendo de un lado a otro en estado de pánico y arrojando su equipo de ping-pong por la borda, "como una especie de sacrificio".

La vista afectó profundamente a Gosper. Antes de esa noche, Gosper había desdeñado el enfoque de la NASA basado en oleadas humanas. Había defendido con firmeza la forma más individualista de elegancia hacker del laboratorio de IA en la programación y en el estilo informático en general. Pero ahora veía cómo el mundo real, cuando tomaba una decisión, podía tener un efecto asombroso. La NASA no había aplicado la ética hacker, pero había hecho algo que el laboratorio, a pesar de todo su espíritu pionero, nunca habría podido hacer. Gosper se dio cuenta de que los hackers del noveno piso se estaban engañando a sí mismos en cierto sentido, trabajando en máquinas de relativamente poca potencia en comparación con las computadoras del futuro, pero aun así tratando de hacerlo todo, de cambiar el mundo allí mismo, en el laboratorio. Y como el estado de la informática aún no había desarrollado máquinas con el poder de cambiar el mundo en general (ciertamente nada que te hiciera retumbar el pecho como lo hizo la operación de la NASA), todo lo que los hackers terminaron haciendo fue crear herramientas para hacer herramientas. Era vergonzoso.

La revelación de Gosper le llevó a creer que los hackers podían cambiar las cosas, sólo hacer que los ordenadores fueran más grandes y más potentes, sin escatimar en gastos. Pero el problema era aún más profundo. Aunque la maestría de los hackers había hecho de la programación informática una búsqueda espiritual, un arte mágico, y aunque la cultura del laboratorio se había desarrollado hasta el punto de convertirse en un estanque tecnológico, algo faltaba esencialmente.

El mundo.

Por mucho que los hackers intentaran crear su propio mundo en el noveno piso, no podían hacerlo. El movimiento de personas clave era inevitable. Y las duras realidades de la financiación golpearon a Tech Square en los años setenta: ARPA, adhiriéndose a la nueva y estricta Enmienda Mansfield aprobada por el Congreso, tuvo que pedir una justificación específica para muchos proyectos informáticos. Los fondos ilimitados para la investigación básica se estaban agotando; ARPA estaba impulsando algunos proyectos favoritos como el reconocimiento de voz (que habría aumentado directamente la capacidad del gobierno para monitorear masivamente las conversaciones telefónicas en el extranjero y en el país). Minsky pensó que la política era "perdedora" y distanció al laboratorio de IA de ella. Pero ya no había suficiente dinero para contratar a alguien que demostrara un talento excepcional para el hacking. Y poco a poco, a medida que el propio MIT se fue afianzando en la formación de estudiantes para los estudios informáticos convencionales, la actitud del Instituto hacia los estudios informáticos cambió un poco de enfoque. El laboratorio de IA empezó a buscar profesores además de investigadores, y los hackers rara vez se interesaban por las molestias burocráticas, las exigencias sociales y la falta de tiempo práctico con las máquinas que conllevaba la enseñanza de cursos.

Greenblatt aún seguía hackeando, al igual que Knight, y algunos hackers más nuevos demostraban ser maestros en el trabajo de sistemas... pero otros se iban, o se iban. Ahora, Bill Gosper se dirigió al Oeste. Se las arregló para quedarse en la nómina del laboratorio de IA, hackeando en el PDP-6 del noveno piso a través de ARPAnet, pero se mudó a California para estudiar el arte de la programación informática con el profesor Donald Knuth en Stanford. Se convirtió en un fijo en Louie's, el mejor restaurante chino de Palo Alto, pero desapareció en acción en Tech Square. Allí, era una presencia voluble en las terminales de ordenador, pero ya no era el centro físico de atención. Estaba sentado sobre una silla y susurraba: “Miren eso”, mientras la terminal 340 latía de forma demencial con nuevas formas de VIDA. Estaba en California y se había comprado un coche.

Con todos estos cambios, algunos de los hackers sintieron que una era estaba terminando. “Antes (en los años sesenta), la actitud era: ‘Aquí están estas nuevas máquinas, veamos qué pueden hacer’”, recordó más tarde el hacker Mike Beeler. “Así que hicimos brazos robóticos, analizamos el lenguaje, hicimos Spacewar... ahora teníamos que justificarlo de acuerdo con los objetivos nacionales. Y (la gente señaló que) algunas cosas que hicimos eran curiosas, pero no relevantes... nos dimos cuenta de que habíamos tenido una situación utópica; toda esta cultura fascinante. Había una cierta cantidad de aislamiento y falta de difusión, de correr la voz. Me preocupaba que todo se fuera a perder”.

No se perdería. Porque había una segunda ola de hackers, un tipo de hacker que no solo vivía según la Ética Hacker, sino que veía la necesidad de difundir ese evangelio lo más ampliamente posible. La forma natural de hacerlo era a través del poder de la computadora, y el momento de hacerlo era ahora. Las computadoras para hacerlo tendrían que ser pequeñas y baratas, haciendo que las minicomputadoras DEC parecieran gigantes de IBM en comparación. Pero Si se utilizaran grandes cantidades de ordenadores pequeños y potentes, el mundo podría cambiar. Había gente que tenía esas visiones, y no eran como Gosper o Greenblatt: eran un tipo diferente de hacker, una segunda generación, más interesada en la proliferación de ordenadores que en hackear aplicaciones místicas de IA. Esta segunda generación estaba formada por hackers de hardware, y la magia que crearían en California se basaría en los cimientos culturales establecidos por los hackers del MIT para difundir el sueño hacker por todo el país.

Parte II. Hackers de Hardware: Norte de California, Los Setentas

Capítulo 8. Rebelión en 2100

La primera terminal pública del proyecto Memoria Comunitaria era una fea máquina en un vestíbulo desordenado en el segundo piso de un edificio destartalado en la ciudad más espaciosa de los Estados Unidos de América: Berkeley, California. Era inevitable que las computadoras llegaran a "el Pueblo" de Berkeley. Todo lo demás lo hizo, desde la comida gourmet hasta el gobierno local. Y si, en agosto de 1973, las computadoras eran generalmente consideradas inhumanas, inflexibles, belicistas y no orgánicas, la imposición de una terminal conectada a uno de esos monstruos orwellianos en una zona normalmente de buenas vibraciones como el vestíbulo exterior de Leopold's Records de la Durant Avenue no era necesariamente una amenaza para el bienestar de nadie. Era otro tipo de ambiente con el que seguir.

En cierto sentido, escandaloso. Una especie de piano aplastado, de la altura de un Fender Rhodes, con un teclado de máquina de escribir en lugar de uno musical. El teclado estaba protegido por una caja de cartón con una placa de vidrio al frente. Para tocar los botones, había que introducir las manos por pequeños agujeros, como si uno se ofreciera a ser encarcelado en una prisión electrónica. Pero las personas que estaban junto a la terminal eran tipos comunes de Berkeley, con el pelo largo y enmarañado, vaqueros, camisetas y un brillo demente en los ojos que si no las conocía bien uno podría confundir con una reacción a estupefacientes. Los que sí los conocían bien se dieron cuenta de que el grupo estaba puesto, con la tecnología. Se estaban divirtiendo como nunca antes, vendiendo el sueño hacker como si fuera la variedad más potente de la sinsemilla del Área de la Bahía.

El nombre del grupo era Community Memory y, según un folleto que distribuyeron, la terminal era "un sistema de comunicación que permite a las personas ponerse en contacto entre sí sobre la base de intereses expresados mutuamente, sin tener que ceder a juicio a terceros". La idea era acelerar el flujo de información en un sistema descentralizado y no burocrático. Una idea nacida de las computadoras, una idea ejecutable sólo por computadoras, en este caso una computadora central XDS-940 de tiempo compartido en el sótano de un almacén en San Francisco. Al abrir un centro de cómputo práctico para permitir que las personas se comuniquen entre sí, se crearía una metáfora viviente, un testimonio de la forma en que la tecnología informática podría usarse como arma en la guerra de guerrillas para el Pueblo contra las burocracias.

Irónicamente, el área pública del segundo piso de Leopold's, la tienda de discos de más onda en East Bay, también era el hogar de la Cartelera de los Músicos, un tablón pared completamente cubierta de anuncios de cantantes vegetarianos que buscaban conciertos, bandas de country jug que buscaban guitarristas dobro, flautistas de Jethro Tull que buscaban compositores con fijaciones similares. El viejo estilo de emparejamiento. La Community Memory alentaba lo nuevo. Podías colocar tu anuncio en la computadora y esperar a que la persona que más lo necesitaba accediera a él de manera instantánea y precisa. Pero los habitantes de Berkeley no tardaron mucho en encontrar otros usos para la terminal:

> DAR CON 1984, DICES
>
> JE, JE, JE... QUÉDATE POR AQUÍ DURANTE OTROS DIEZ AÑOS
>
> ESCUCHA A ALVIN LEE
>
> RÉGATE EL PELO DE OTRA MANERA

TÓMATE UNA ASPIRINA

HAGAMOS UN ESFUERZO CONJUNTO

ALÉJATE

MANTEN LA NARIZ LIMPIA

EN CASA (EN EL CAMPO DE APLICACIÓN)

DEJA DE PATEAR TUS CORAZONES, MÍRAME, SIENTEME

EE. UU., LÁRGATE DE WASHINGTON

LIBERA LA CARRERA 500 DE INDIANÁPOLIS

LEVÁNTATE Y ALEJATE

QUEDATE EN EL CAMINO

VOLTEATE

ENDEREZCA

DEJE QUE UNA SONRISA SEA SU PARAGUAS

. . . Y . . .

ANTES DE QUE TE DES CUENTA {}{}{}{}{}{}{}{}{}{}{}

1984

TE

ENCONTRARÁ!

Y SERÁ JUSTO . . .

PALABRAS CLAVE: 1894 BENWAY TLALCLATLAN INTERZONE

20-2-74

Fue una explosión, una revolución, un demencial piña contra el establishment, encabezado por un usuario demente (el usuarísimo, ligado al Pueblo) que se hacía llamar Doctor Benway en homenaje a un personaje sádicamente pervertido de Naked Lunch de Burroughs. Este tipo, Benway, estaba llevando las cosas más al límite de lo que incluso los más radicalizados informáticos de Community Memory habían sospechado que irían, y los radicales informáticos estaban encantados.

Nadie estaba más feliz que Lee Felsenstein. Fue uno de los fundadores de Community Memory y, aunque no era necesariamente su miembro más influyente, era un símbolo del movimiento que estaba llevando la ética hacker a las calles. En la década siguiente, Lee Felsenstein promovería una

versión del sueño hacker que - de haberlo sabido - horrorizaría a Greenblatt y a los trabajadores de IA de Tech Square con su ingenuidad tecnológica, su base política y su voluntad de difundir el evangelio de la informática a través - nada menos - que el mercado. Pero Lee Felsenstein sentía que no debía nada a esa primera generación de hackers. Era una nueva raza, un hacker del hardware y combatiente del pueblo. Su objetivo era sacar a los ordenadores de las marfilescas torres protegidas del IA, sacarlos de las profundas mazmorras de los departamentos de contabilidad empresarial, y permitir que el Pueblo cobrara autoconeciencia mediante el Imperativo de la Manos a la Obra. A él en esta lucha se le unirían otros, que simplemente hackeaban hardware - no con propósito político alguno, sino por el mero placer de la militancia en sí misma; estas personas desarrollarían las máquinas y los accesorios a través de los cuales se llevaría a cabo la práctica. El uso de la informática militante se generalizaría tanto que el concepto mismo de la misma cambiaría: sería más fácil para todos sentir la magia. Lee Felsenstein estaría tan cerca como nadie de ser un comandante para estas tropas rabiosamente anarquistas; pero ahora, como miembro de Community Memory, formaba parte de la lucha colectivo para dar los primeros pasos en una batalla trascendental que los hackers del MIT nunca habían considerado que valiera la pena librar: difundir la ética hacker llevando las computadoras al Pueblo.

Era la visión de Lee Felsenstein del sueño hacker, y sentía que había pagado sus cuotas al adquirirla.

• • • • • • •

La infancia de Lee Felsenstein bien podría haberlo calificado para un puesto entre la élite hacker en el noveno piso del Tech Square. Se trataba de la misma fijación con la electrónica, algo que se apoderó de él de forma tan extraña que desafiaba toda explicación racional. Sin embargo, Lee Felsenstein intentaría más tarde dar una explicación racional a su amor por la electrónica. En sus reconstrucciones de sus primeros años - reconstrucciones moldeadas por años de terapia -, atribuiría su fascinación por la tecnología a una compleja amalgama de impulsos psicológicos, emocionales y de supervivencia, así como al simple y llano Imperativo del Manos a la Obra. Y sus peculiares circunstancias garantizaron que se convertiría en un hacker de un tipo diferente al de Kotok, Silver, Gosper o Greenblatt.

Nacido en 1945, Lee creció en el barrio Strawberry Mansion de Filadelfia, un barrio de casas adosadas habitado por inmigrantes judíos de primera y segunda generación. Su madre era hija de un ingeniero que había inventado un importante inyector de combustible diésel, y su padre - un artista comercial - había trabajado en una fábrica de locomotoras. Más tarde - en una semblanza autobiográfica inédita - Lee escribiría que su padre Jake “era un modernista que creía en la ‘perfectibilidad’ del hombre y la máquina como modelo de la sociedad humana. Cuando jugaba con sus hijos, solía imitar una locomotora de vapor, como otros hombres imitaban a los animales”.

La vida familiar de Lee no era feliz. La tensión familiar era muy alta; había guerras entre hermanos entre Lee, su hermano Joe (tres años mayor) y una prima de la edad de Lee que fue adoptada como hermana de los chicos. Las aventuras políticas de su padre Jake como miembro del Partido Comunista habían terminado a mediados de los años cincuenta, cuando las luchas internas llevaron a Jake a perder su puesto como puntero del distrito, pero la política era central para la familia. Lee participó en marchas en Washington, D.C. a la edad de doce y trece años, y una vez hizo un piquete en Woolworth’s en una de las primeras manifestaciones por los derechos civiles. Pero cuando las cosas en casa se pusieron demasiado intensas para él, se retiraba a un taller en el sótano lleno de

piezas electrónicas de televisores y radios abandonados. Más tarde, Felsenstein llamaría a ese taller su Monasterio, un refugio donde hizo votos de lealtad a la tecnología.

Era un lugar donde la ineludible superioridad física y académica de su hermano no se extendía. Lee Felsenstein despuntaba una habilidad con la electrónica que le permitió superar a su hermano por primera vez. Era un poder que casi temía desplegar: construía cosas pero nunca se atrevía a encenderlas, temiendo un fallo que confirmara la afirmación de su hermano de que "esas cosas nunca van a funcionar". Así que construía otra cosa en su lugar.

Le encantaba la idea de la electrónica. Llenó la tapa de su cuaderno de sexto grado con diagramas eléctricos. Iba a la sucursal de su barrio de la Biblioteca Libre de Filadelfia y hojeaba las páginas del Manual del Radioaficionado. Lo que más le emocionaba era un manual de instrucciones de la Compañía Heath para construir un receptor de onda corta. La Compañía Heath se especializaba en proyectos electrónicos para hacer uno mismo, y este manual en particular tenía diagramas muy detallados de cables y conexiones. Al comparar las piezas reales de ese proyecto de cinco tubos con el diagrama perfecto, con sus octógonos conectados a otros octógonos, Lee vio la conexión... esta línea del esquema representaba ese pin en el zócalo del tubo. Le produjo una emoción casi sensual, esta conexión de su mundo de fantasía electrónica con la realidad. Llevaba el manual a todas partes; un peregrino que lleva un libro de oraciones. Pronto estaba completando proyectos y se vio reivindicado cuando a los trece años ganó un premio por su modelo de satélite espacial, cuyo nombre era un homenaje a la Madre Rusia, el Felsnik.

Pero aunque se estaba dando cuenta de sí mismo de una manera que nunca antes lo había hecho, cada uno de los nuevos productos de Lee era una aventura de paranoia, ya que temía no poder conseguir la pieza que lo hiciera funcionar. "Siempre veía estos artículos [de Popular Mechanics] que decían: 'Caramba, si tienes este transistor podrías hacer una radio normal que siempre quisiste, y hablar con tus amigos y hacer nuevos amigos'... pero nunca pude conseguir ese componente y realmente no sabía cómo conseguirlo, o no podía conseguir el dinero para obtenerlo". Se imaginó la voz burlona de su hermano, etiquetándolo de fracasado.

Cuando Lee era estudiante de primer año en Central High, la escuela secundaria académica especial para varones de Filadelfia, su hermano Joe, un estudiante de último año, recultó a Lee para convertirlo en el ingeniero jefe del incipiente Club de Computación de la escuela, le mostró a Lee un diagrama de unos circuitos flip-flops obsoletas y desafió a su hermano menor a construirlos. Lee estaba demasiado aterrorizado para decir que no, e intentó sin éxito completar el proyecto. El esfuerzo lo hizo desconfiar de las computadoras durante una década después.

Pero la escuela secundaria elevó a Lee: participó en grupos políticos, trabajó un poco en el ciclotrón de la escuela y leyó algunas cosas importantes, en particular algunas novelas de Robert Heinlein.

El adolescente judío de complexión delgada y anteojos se identificó de alguna manera con los protagonistas futuristas, en particular con el joven soldado virginal de *Revolt in 2100*. El escenario de la novela es una dictadura del siglo XXI, donde una fuerza devota e idealista en la clandestinidad devota e idealista conspira para luchar contra las fuerzas de *El Profeta*, un matón orwelliano omnipotente apoyado por masas irreflexivas que lo adoran. El protagonista se topa con pruebas de la hipocresía del Profeta y - obligado a elegir entre el bien y el mal - toma la drástica decisión de unirse a la Cábala Revolucionaria, que le proporciona las enseñanzas necesarias para estimular su imaginación.

> Por primera vez en mi vida leía cosas que no habían sido aprobadas por los censores del Profeta, y el impacto en mi mente fue devastador. A veces miraba por encima del hombro para ver quién me observaba, asustado a pesar de mí mismo. Empecé a sentir vagamente que el secreto es la piedra angular de toda tiranía.
>
> (de Rebelión en 2100)

Al leer esa novela, y más tarde *Extraño en tierra extraña*, en la que el protagonista extraterrestre de Heinlein se convierte en líder de un grupo espiritual que tiene un profundo efecto en la sociedad, Lee Felsenstein comenzó a ver su propia vida como algo parecido a una novela de ciencia ficción. Los libros - afirmaría más tarde - le dieron coraje para soñar en grande, probar proyectos arriesgados y superar sus propios conflictos emocionales. La gran lucha no era tanto interna como amplia: era la elección entre el bien y el mal. Lee, que se tomó a pecho esa idea romántica, se veía a sí mismo como la persona común y corriente con potencial que - presa de las circunstancias - elige el difícil camino de ponerse del lado del bien y se embarca en una larga odisea para derrocar al mal.

No pasó mucho tiempo antes de que Lee pudiera aplicar esta metáfora en la realidad. Después de graduarse, fue a la Universidad de California en Berkeley para matricularse en Ingeniería Eléctrica. No pudo obtener una beca. Su primer año en la universidad no fue comparable al de un hacker típico del MIT: más o menos siguió la línea, y no logró calificar para una beca por una fracción de punto de calificación. Pero consiguió lo que parecía ser igual de bueno: una asignación de estudio y trabajo en el Centro de Investigación de Vuelo de la NASA en la Base Aérea Edwards, al borde del desierto de Mojave. Para Lee, era la entrada al Paraíso: el idioma que hablaban allí era la electrónica, la electrónica de cohetes, y los esquemas que había estudiado ahora se transfigurarían en material de ciencia ficción hecho realidad. Se deleitaba con la hermandad de los ingenieros, le encantaba llevar corbata, salir de una oficina y ver hileras ordenadas de otras oficinas y dispensadores de agua. Heinlein quedó olvidado: Lee se conformaba, era un ingeniero hecho con un molde. Delirantemente feliz al servicio del Profeta. Luego, después de dos meses de ese "séptimo cielo", como lo llamó más tarde, fue convocado a una reunión con un oficial de seguridad. El oficial parecía incómodo. Estaba acompañado por un testigo de procedimientos. El oficial tomó notas e hizo que Lee firmara cada página a medida que la terminaba. También tenía el formulario que Lee había llenado al ingresar a Edwards, el Formulario de Seguridad 398. El oficial le preguntaba constantemente a Lee si conocía a alguien que fuera miembro del Partido Comunista. Y Lee seguía diciendo que no. Finalmente, preguntó, con voz suave: "¿No entiendes que tus padres eran comunistas?". Lee nunca se lo había dicho. Había asumido que "comunista" era solo un término –anticomunista– que la gente lanzaba a los liberales activistas como sus padres. Su hermano lo sabía (¡su hermano se llamaba Stalin!), pero a Lee no se lo habían dicho. Había sido perfectamente honesto cuando rellenó el Formulario 398 con un claro "no" en la línea que preguntaba si conocía a algún comunista conocido.

"Así que allí estaba yo, expulsado del Paraíso", diría Lee más tarde, "y el jefe de seguridad me dijo: 'Mantén tu nariz limpia durante un par de años más, no tendrás ningún problema para volver a entrar'. Ahora bien, siempre me había estado preparando para que me abandonaran, siempre había esperado que me abandonaran. De repente, me abandonaron. Literalmente, me arrojaron al desierto. ¡Ahí está el desierto de Mojave, por el amor de Dios!".

La noche del 14 de octubre de 1964, Lee Felsenstein, el ingeniero fracasado, tomó un tren de regreso a Berkeley. Lee había oído informes de radio sobre manifestaciones estudiantiles que habían

comenzado allí dos semanas antes; las había descartado como una versión moderna de las legendarias redadas de bragas que habían ocurrido en 1952. Pero a su regreso encontró a toda la comunidad ardorosamente viva con el Movimiento de la Libertad de Expresión. “El secreto es la piedra angular de toda tiranía”, decía el protagonista de Rebelion en 2100 de Heinlein, siendo ya no sólo grito de la revolución de Berkeley, sino también el de la ética hacker. Lee Felsenstein dio el salto: se unió a la Cábala. Pero fusionaría su fervor con su propio talento particular. Utilizaría la tecnología para sobrealimentar la revuelta.

Como poseía una grabadora, fue a Press Central - el centro de medios del movimiento - y ofreció sus talentos como técnico de audio. Hizo un poco de todo: mimeografió, hizo trabajos de mierda. Se inspiró en la estructura descentralizada del Movimiento por la Libertad de Expresión. El 2 de diciembre, cuando más de ochocientos estudiantes ocuparon el Sproul Hall, Lee estaba allí con su grabadora. Fue arrestado, por supuesto, pero la administración dio marcha atrás en los temas. La batalla estaba ganada, pero la guerra apenas comenzaba.

Durante los siguientes años, Lee equilibró las existencias aparentemente incompatibles de un activista social y un ingeniero socialmente solitario. No muchos en el movimiento tenían tanta inclinación técnica; la tecnología y especialmente las computadoras eran percibidas como fuerzas malignas. Lee trabajó furiosamente para organizar a la gente en su dormitorio cooperativo, el Oxford Hall, el más militante de todo el campus. Editó el órgano del dormitorio. Pero también estaba aprendiendo más sobre electrónica, jugando con circuitos, sumergiéndose en el hermandad lógica de capacitores y diodos. En la medida de lo posible, fusionó ambas actividades: diseñó, por caso, una herramienta que era una combinación de megáfono y garrote para defenderse de los policías. Pero a diferencia de muchos en el movimiento que también estaban profundamente involucrados en la actividad social desenfrenadamente radicalizada de Berkeley, Lee rehuía el contacto humano cercano, especialmente con las mujeres. Lee, una figura sucia con ropa de trabajo, vivía conscientemente a la altura del estereotipo del ingeniero mugroso. No se bañaba con regularidad y se lavaba el pelo corto, poco de moda, quizás una vez al mes. No se drogaba. No practicaba sexo, y mucho menos el sexo libre que acompañaba la libertad de expresión. "Tenía miedo de las mujeres y no tenía forma de tratar con ellas", explicó más tarde. "Tenía una especie de prohibición en mi personalidad contra la diversión. No se me permitía divertirme. La diversión estaba en mi trabajo... Era como si mi manera de afirmar mi potencia fuera poder construir cosas que funcionaran y que gustaran a otras personas”.

Lee abandonó Berkeley en 1967 y comenzó a alternar entre trabajos de electrónica y trabajo en el movimiento. En 1968, se unió al periódico clandestino Berkeley Barb como “editor militante”. Uniéndose a otros escritores como Sergeant Pepper y Jefferson Fuck Poland, Lee escribió una serie de artículos evaluando las manifestaciones, no sobre la base de consignas, sino de organización, estructura, conformación a un sistema elegante. En uno de sus primeros artículos, en marzo de 1968, Lee habló de una próxima manifestación para la Semana del Stop-the-Draft, señalando el resultado probable de una planificación insuficiente y disputas entre los organizadores: “La actividad será a medias, caótica y como todas las demás manifestaciones. Los burócratas del movimiento parecen no darse cuenta de que en el mundo real la acción se lleva a cabo no en virtud de sutilezas ideológicas, sino con tiempo y recursos físicos. . . es mi responsabilidad como técnico no sólo criticar sino hacer sugerencias. . .”

Y sí hizo sugerencias. Insistió en que las manifestaciones debían ejecutarse con tanta limpieza como los circuitos lógicos definidos por los esquemas precisos que todavía veneraba. Elogió a los

manifestantes cuando destrozaron “las correctas ventanas” (bancos, no pequeñas empresas). Abogó por el ataque sólo para atraer al enemigo. Calificó de “refrescante” el bombardeo de una oficina de reclutamiento. Su columna titulada “Consejos domésticos del editor militante” aconsejaba: “Recuerde girar la gelinita almacenada cada dos semanas cuando hace calor. Esto evitará que los nitrogliceratos se peguen”.

El protagonista de Heinlein en Rebelión en 2100 dijo: “La revolución no va acompañada de un puñado de conspiradores susurrando alrededor de una vela que se apaga en una ruina desierta. Requiere innumerables suministros, maquinaria moderna y armas modernas. . . y debe haber lealtad. . . y una organización de personal superlativa”. En 1968, Lee Felsenstein escribió: “La revolución es mucho más que una pelea callejera al azar. Hace falta organización, dinero, determinación tenaz y voluntad de aceptar y construir sobre los desastres pasados”.

Felsenstein tuvo su efecto. Durante el juicio de los Siete de Oakland, el abogado defensor Malcolm Burnstein dijo: “No deberíamos tener a estos acusados aquí... debería haber sido Lee Felsenstein”.

• • • • • • • •

En el verano de 1968, Lee Felsenstein puso un anuncio en el Barb. El anuncio en sí era poco explícito: “hombre del Renacimiento, ingeniero y revolucionario, buscando conversación”. No mucho después, una mujer llamada Jude Milhon encontró el anuncio. En comparación con otros comentarios insinuantes de las últimas páginas del Barb (“¡SOLO CHICAS! Anhelo tus pies”), parecía que provenía de un hombre decente, pensó. Era lo que Jude necesitaba en ese año tumultuoso: una veterana del movimiento por los derechos civiles y una activista de larga trayectoria, había quedado aturdida por los acontecimientos políticos y sociales del ‘68. El mundo parecía desmoronarse.

Jude no era sólo una activista, sino también una programadora de ordenadores. Había sido amiga íntima de un hombre llamado Efrem Lipkin, que también estaba en el movimiento, y era un mago de la informática que le enviaba acertijos para entretenerse; ella no dormía hasta que los resolvía. Aprendió a programar y le pareció una delicia, aunque nunca entendió por qué los hackers lo encontraban obsesivamente absorbente. Efrem iba a venir del Este para reunirse con ella en la Costa en unos meses, pero mientras tanto se sentía lo bastante sola como para ponerse en contacto con el hombre que escribió el anuncio en el Barb.

Jude, una mujer rubia, delgada y valiente, de ojos azules firmes, inmediatamente consideró a Lee como un “tecnorraro por excelencia”, pero sólo por su propia creación. Casi sin darse cuenta, gracias a su compañía, y en particular a su constante franqueza, perfeccionada en incontables sesiones de autoevaluación en varios colectivos, Jude comenzó el largo proceso de extraer la personalidad de Lee Felsenstein. Su amistad era más profunda que una relación de noviazgo y continuó mucho después de que su amiga Efrem llegara de la Costa Este. Lee se hizo amigo de Efrem, que no solo era activista sino también hacker informático. Efrem no compartía la creencia de Lee de que la tecnología podía ayudar al mundo; sin embargo, la cautela de Lee durante una década sobre las computadoras estaba llegando a su fin. Porque, en 1971, Lee tenía una nueva compañera de habitación: una computadora XDS-940.

Pertenecía a un grupo llamado Resource One, parte del Proyecto Uno, un grupo de superficie del Área de la Bahía que fomentaba el activismo comunitario y los programas humanísticos. “One” había sido iniciado por un arquitecto-ingeniero que quería dar a los profesionales desempleados algo útil que hacer con sus habilidades, ayudar a la comunidad y comenzar a disipar el “aura de

elitismo, e incluso misticismo, que rodea al mundo de la tecnología". Entre los proyectos que se desarrollaban en el almacén de cinco pisos de color amarillo mostaza de One, en una zona industrial de San Francisco, se encontraba el colectivo Resource One, formado por personas "que creen que las herramientas tecnológicas pueden ser herramientas de cambio social cuando las controla el Pueblo". La gente de Resource One había convencido a la Corporación Transamerica para que prestara al grupo un ordenador XDS-940 de tiempo compartido sin uso, de modo que One pudiera empezar a reunir listas de correo alternativas y a poner en marcha su programa de educación informática, proyectos de investigación económica y "desmitificación para el público en general".

El ordenador era un gigante descomunal, una máquina de 800.000 dólares que ya estaba obsoleta. Ocupaba una habitación y requería una unidad de aire acondicionado de veintitrés toneladas. Necesitaba un encargado de sistemas a tiempo completo para ponerla en marcha. Resource One necesitaba un hacker, y Lee Felsenstein parecía una elección lógica.

El software de sistemas lo instaló un hacker de Xerox PARC (Palo Alto Research Center) que había escrito el sistema de tiempo compartido original para el 940 en Berkeley. Era un tal Peter Deutsch de pelo largo y barba, el mismo Peter Deutsch que doce años antes y a los doce años había mirado por encima de la consola del TX-0. Graduado de Berkeley, había logrado combinar el estilo de vida californiano de toda la Tierra con el intenso trabajo de hackers en PARC.

Pero era Lee quien era el cuidador de la máquina. En su continua mitificación de su vida como novela de ciencia ficción, consideró este período como una reinmersión en el papel asocial de una persona cuyo mejor amigo era una máquina, un esteta tecnológico que se sacrificaba al servicio de la Cábala. El monasterio esta vez sería el sótano del almacén de Resource One; por treinta dólares al mes alquilaba una habitación. Estaba por debajo del nivel del alcantarillado, no tenía agua corriente, estaba sucio. Para Lee era perfecto: "Iba a ser un siervo invisible. Parte de esta máquina".

Pero Resource One le quedó chico a Lee, que estaba muy por delante del grupo al darse cuenta de que los usos sociales de la tecnología dependerían del ejercicio de algo parecido a la ética hacker. Los demás miembros del grupo no crecieron ansiando la tecnología práctica... su conexión con ella no era visceral sino intelectual. Como resultado, discutían sobre cómo se debería utilizar la máquina en lugar de armarla y usarla. Eso volvía loco a Lee.

Lee explicó más tarde: "Éramos mojigatos, éramos estetas intolerables. Cualquiera que quisiera utilizar la máquina tenía que venir a exponer su caso ante nuestra reunión. Parecía que había que implorar para poder usarla". Lee quería cambiar la perspectiva del grupo a una más abierta, más parecida a la de los hackers, pero no tenía el coraje para hacer el esfuerzo social: su autoestima había llegado a su punto más bajo. Rara vez tenía el coraje de aventurarse a salir del edificio para enfrentarse al mundo; cuando lo hacía, observaba con tristeza que los vagabundos del distrito Tenderloin parecían más limpios, más prósperos que él. Otras personas del colectivo intentaron abrirlo; Una vez, durante una reunión, pidieron prestada una cámara de televisión a un colectivo de vídeos que había en el piso de arriba y, cada vez que se oían risas en el grupo, enfocaban a Lee, invariablemente con cara de póquer. Al mirar la cinta después, pudo ver en qué se estaba convirtiendo: en un desalmado. "Sentí que no podía permitirme tener corazón", dijo. "Yo ya veía que esto iba a pasar, pero los estaba alejando".

Después de esa experiencia, trató de volverse más activo en influenciar al grupo. Se enfrentó a un fanático de Goldbricker que se pasaba la mayor parte del día bebiendo café lentamente. "¿Qué has estado haciendo?", le preguntó Felsenstein. El tipo empezó a hablar de ideas vagas y Lee dijo: "No

te estoy preguntando qué quieres hacer, te estoy preguntando qué has hecho". Pero pronto se dio cuenta de que llamar la atención a la gente por sus tonterías era inútil: como una máquina ineficiente, la arquitectura del grupo en sí misma era defectuosa. Era una burocracia. Y el hacker que había en Lee no podía soportar eso. Afortunadamente, en esa época, en la primavera de 1973, Efrem Lipkin llegó a Resource One para rescatar a Lee Felsenstein y hacer despegar Community Memory.

Efrem Lipkin era el tipo de persona que podía mirarte con los ojos entornados y una cara alargada y semítica, y sin decir una palabra, hacerte saber que el mundo tenía lamentables defectos y que tú no eras la excepción. Era el aire de un purista que nunca podría cumplir con sus propios estándares exigentes. Efrem acababa de regresar de Boston, donde había estado en la nómina de una empresa de consultoría informática. La empresa se había enfrascado en contratos relacionados con el ejército, y Efrem había dejado de ir a trabajar. El programador idealista no informó a su empleador; simplemente dejó de trabajar, con la esperanza de que el proyecto se detuviera por su falta de participación. Después de nueve meses, durante los cuales la empresa asumió que estaba trabajando duro, quedó claro que no había programa alguno, y el presidente de la empresa fue a su apartamento infestado de cucarachas de Cambridge y le preguntó: "¿Por qué hiciste esto?". Le dijo a Efrem que había creado la empresa después de que Martin Luther King muriera, para hacer el bien. Insistió en que los proyectos que asumiera mantendrían al país fuerte contra la amenaza tecnológica japonesa. Efrem sólo vio que la empresa con la que tenían contrato había estado involucrada en armas antipersonales durante la guerra. ¿Cómo podía trabajar para esa empresa? ¿Cómo se podía esperar que hiciera algún trabajo informático, teniendo en cuenta que sus usos con demasiada frecuencia son perjudiciales?

Era una pregunta que había atormentado a Efrem Lipkin durante años.

Efrem Lipkin había sido un hacker desde la escuela secundaria. Su afinidad por la máquina fue instantánea y consideraba que la programación era "la actividad incorpórea definitiva: me olvidaba de hablar inglés. Mi mente trabaja en formas informáticas". Pero - a diferencia de algunos de sus compañeros en un programa especial para informáticos de la ciudad de Nueva York - Efrem también consideraba que su asombroso talento para la computadora era una maldición. Al igual que Lee, provenía de una familia de política virulentamente izquierdista y, además de deslumbrar a sus profesores de matemáticas, lo habían expulsado de clase de Historia por no saludar a la bandera y por llamar mentiroso al profesor. A diferencia de Lee, que buscaba combinar la tecnología y la política, Efrem las veía como opuestas, una actitud que lo mantenía en constante agitación.

"Amo las computadoras y odio lo que las computadoras pueden hacer", diría más tarde. Cuando iba al instituto, consideraba que las aplicaciones comerciales de los grandes mainframes (enviar facturas y cosas así) eran simplemente poco interesantes. Pero cuando comenzó la guerra de Vietnam, comenzó a ver sus juguetes favoritos como instrumentos de destrucción. Vivió en Cambridge durante un tiempo y un día se aventuró a subir al noveno piso de Tech Square. Vio el PDP-6, vio la pequeña cabeza de playa perfecta de la ética hacker que se había establecido allí, vio el virtuosismo y la pasión concentrados, pero solo podía pensar en la fuente de financiación y las aplicaciones finales de esta magia desenfrenada. "Me enojé tanto que comencé a llorar", dijo más tarde. "Porque esta gente me había robado la profesión. Hicieron imposible ser un informático. Se vendieron. Se vendieron a los usos militares, los usos malignos, de la tecnología. Eran una subsidiaria de propiedad absoluta del Departamento de Defensa".

Así que Efrem se fue a California, luego de nuevo a la Costa Este, y luego de nuevo a California. Le llevó un tiempo ver cómo se podían usar las computadoras para el bien social, y cada vez que vislumbraba las posibilidades sospechaba que había una traición. Un proyecto interesante en el que había participado era el juego del Mundo. Un grupo de programadores, filósofos e ingenieros de California construyeron una simulación del mundo. Se basaba en una idea de Buckminster Fuller, donde se podían probar todo tipo de cambios y ver su efecto en el mundo. Durante días, la gente iba de un lado a otro sugiriendo cosas y ejecutando el juego en la computadora. No se obtuvieron muchas sugerencias sobre cómo manejar el mundo, pero mucha gente conoció a otras personas con puntos de vista similares.

Poco después, Efrem se topó con Resource One, con Lee sumido en sus entrañas. Pensó que era una tontería. Estaba esta gran instalación con una computadora y un poco de software para bases de datos comunitarias y una centralita, pero el grupo no estaba haciendo todo lo que podía. ¿Por qué no llevar esa gran infraestructura a las calles? Efrem empezó a entusiasmarse con la idea y, quizás por primera vez en su vida, vio cómo las computadoras realmente podrían ser propicias para el bien social. Hizo que Lee se lo pensara y trajo a otras personas que había conocido en el juego World.

La idea era formar un brazo de Resource One denominado Community Memory. Computadoras sueltas en nuestras calles, liberando al Pueblo para que hicieran sus propias conexiones. Felsenstein presionó a la gente de Resource One para que pagaran una oficina en Berkeley que también podría ser su apartamento. Entonces la facción Community Memory se mudó al otro lado de la bahía, a Berkeley, para poner en marcha el sistema. Y Lee se sintió liberado de su institucionalización autoimpuesta. Formaba parte de un grupo imbuido del espíritu hacker, dispuesto a hacer algo con las computadoras, todos entusiasmados con la idea de que el acceso a las terminales uniría a las personas con una eficiencia inaudita y, en última instancia, cambiaría el mundo.

• • • • • • •

Community Memory no era el único intento en curso de llevar las computadoras a la gente. En toda la zona de la Bahía, los ingenieros y programadores que amaban las computadoras y se habían politizado durante el movimiento contra la guerra de Vietnam, estaban pensando en combinar sus dos actividades. Un lugar en particular parecía combinar una irreverencia contracultural relajada con un impulso evangélico para exponer al Pueblo, especialmente a los niños, a las computadoras. Se trataba de la People’s Computer Company. Fiel al estilo caprichoso de su fundador, la People’s Computer Company no era realmente una compañía. La *orga*, con nombre inapropiado si alguna vez existió uno, publicó un órgano con ese nombre, pero lo único que realmente produjo fue un intenso sentimiento por la computación por sí misma. Lee Felsenstein asistía con frecuencia a las cenas compartidas de los miércoles por la noche de la PCC, que proporcionaban un punto de encuentro común para los contraculturistas informáticos de la zona de la Bahía, así como la oportunidad de ver a Bob Albrecht intentar - por enésima vez - enseñar a todo el mundo la danza folclórica griega.

Bob Albrecht era el visionario detrás de la People’s Computer Company. Era un hombre - diría más tarde Lee Felsenstein - para quien «acercar a un niño a una computadora era como abusar de un niño». Como abusar de un niño, es decir, para un pederasta obsesivo.

En la primavera de 1962, Bob Albrecht había entrado en un aula y había tenido una experiencia que cambiaría su vida. A Albrecht - que entonces trabajaba para la Control Data Company como analista de aplicaciones senior - le habían pedido que hablara en el club de matemáticas de la escuela

secundaria George Washington de Denver, un grupo de judíos triunfadores, aunque bien educados. Albrecht, un hombre grande con corbata de clip, nariz carnosa y ojos azul marino que podían brillar con fuerza creativa o caer como los de un basset detrás de sus lentes de montura cuadrada, dio su pequeña charla sobre computadoras y preguntó casualmente si alguno de los treinta y dos estudiantes querría aprender a programar una computadora. Treinta y dos manos se agitaron en el aire.

Albrecht nunca había visto una respuesta como esa cuando enseñaba FORTRAN correctivo, su "curso de un día para gente que había ido a la escuela de IBM y no había aprendido nada", como dijo más tarde. Albrecht no podía entender cómo IBM podía haber dado clases a esa gente y no haberles dejado hacer nada. Ya entonces sabía que el juego se basaba en la práctica, como siempre había sido desde que empezó con los ordenadores en 1955 en la división aeronáutica de Honeywell. A lo largo de una sucesión de empleos, se había sentido constantemente frustrado por las burocracias. Bob Albrecht prefería un entorno flexible; era un estudioso de la serendipia en cuanto a estilo de vida y perspectivas. Llevaba el pelo corto, llevaba la camisa abotonada y su perfil familiar (esposa, tres hijos, un perro) no tenía nada de excepcional. Sin embargo, en el fondo, Bob Albrecht era un bailarín griego, ansioso por sacar a relucir el ouzo y el bouzouki. Baile griego, licor y mainframes: esos eran los elementos para Bob Albrecht. Y se sorprendió al descubrir cuán ansiosos estaban los estudiantes de secundaria por entregarse a este último placer, el más seductor de los tres.

Comenzó a dar clases nocturnas para los estudiantes de la oficina de Control Data. Albrecht descubrió que el deleite de los jóvenes al aprender a tomar el control de la computadora Control Data 160A era intenso, adictivo, visceral. Estaba mostrando una nueva forma de vida a los niños. Estaba trasvasando poder.

Albrecht no se dio cuenta entonces, pero estaba difundiendo el evangelio de la ética hacker, ya que los estudiantes intercambiaban programas y compartían técnicas. Comenzó a imaginar un mundo donde las computadoras abrirían el camino hacia un nuevo estilo de vida liberador. Si tan solo estuvieran al alcance... Poco a poco, comenzó a ver la misión de su vida: difundir esta magia por todo el terrunio.

Albrecht contrató a cuatro de sus mejores estudiantes para que programaran por alrededor de un dólar la hora. Se sentaban allí en sus escritorios, escribiendo alegremente programas para resolver funciones cuadráticas. La máquina aceptaba sus tarajetas y las masticaba mientras ellos observaban felices. Luego, Albrecht les pidió a estos estudiantes destacados que enseñaran a sus compañeros. "Su idea era hacer que nos multiplicáramos lo más rápido posible", dijo más tarde uno de los integrantes del grupo, un chico pelirrojo llamado Bob Kahn.

Albrecht usó a los cuatro como "pregoneros" para un "programa de medicina" en su escuela secundaria. Los estudiantes estaban completamente a cargo. Veinte clases de matemáticas participaron en el programa, para el cual Albrecht había convencido a sus empleadores de desprenderse de la 160A y una Flexowriter durante una semana. Después de mostrarles a las clases algunos trucos de matemáticas, le preguntaron a Kahn si la computadora podía hacer los ejercicios que estaban al final de un texto de matemáticas, y procedió a hacer la tarea del día, usando la Flexowriter para cortar un formulario mimeografiado para que cada estudiante tuviera una copia. Sesenta estudiantes se sintieron motivados por el programa de medicina para inscribirse en clases de computación; y cuando Albrecht llevó el programa de medicina a otras escuelas secundarias, la respuesta fue igualmente entusiasta. Pronto Albrecht presentó triunfalmente su programa de

medicina en la Conferencia Nacional de Computación, donde sus jóvenes prodigio asombraron a los sumos sacerdotes de la industria. Nosotros no hacemos eso, le dijeron a Albrecht.

Él se sacudió de alegría. Lo haría. Convenció a Control Data para que le permitiera llevar el programa de medicina a todo el país y trasladó su base a la sede de CDC en Minnesota. Fue allí donde alguien le mostró BASIC, el lenguaje de programación desarrollado por John Kemeny de Dartmouth para dar cabida, escribió Kemeny, "a la posibilidad de que millones de personas escribieran sus propios programas informáticos... Aprovechando años de experiencia con FORTRAN, diseñamos un nuevo lenguaje que era especialmente fácil de aprender para el profano [y] que facilitaba la comunicación entre el hombre y la máquina". Albrecht decidió inmediatamente que BASIC era lo que buscaba y que FORTRAN estaba muerto. BASIC era interactivo, de modo que la gente ávida de usar la computadora obtendría una respuesta instantánea de la máquina (FORTRAN estaba orientado al procesamiento por lotes). Utilizaba palabras similares al inglés como INPUT, THEN y GOTO, de modo que era más fácil de aprender. Y tenía un generador de números aleatorios incorporado, de modo que los niños podían usarlo para escribir juegos rápidamente. Albrecht sabía incluso entonces que los juegos proporcionarían el aroma seductor que atraería a los niños a la programación y al hackerismo. Albrecht se convirtió en un profeta de BASIC y finalmente cofundó un grupo llamado SHAFT (Sociedad para ayudar a abolir la enseñanza de FORTRAN).

A medida que se fue involucrando más en los aspectos misioneros de su trabajo, finalmente salió a la superficie el Bob Albrecht que bullía bajo su apariencia formal. Cuando los años sesenta estaban en pleno apogeo, Albrecht se fue a California, divorciado, con el pelo largo, ojos brillantes y una cabeza llena de ideas radicales sobre cómo exponer a los niños a las computadoras. Vivía en la cima de Lombard Street (la colina más alta y sinuosa de San Francisco) y mendigaba o pedía prestadas computadoras para su práctica evangelizadora. Los martes por la noche abría su apartamento para sesiones que combinaban cata de vinos, baile griego y programación de computadoras. Estuvo involucrado en la influyente Universidad Libre de Midpeninsula, una encarnación de la actitud de "haz lo que quieras" de la zona, que atraía a personas como Baba Ram Dass, Timothy Leary y el ex sabio de la inteligencia artificial del MIT, "el tío" John McCarthy. Albrecht participó en la creación de la "división de educación informática" de la fundación sin fines de lucro llamada Instituto Portola, que más tarde engendró el *Catálogo de la Tierra Entera*. Conoció a un profesor de la escuela secundaria Woodside de la península, llamado LeRoy Finkel, que compartía su entusiasmo por enseñar informática a los niños; con Finkel fundó una empresa de publicación de libros de informática llamada Dymax, en honor a la palabra registrada de Buckminster Fuller, "dymaxion", que combinaba dinamismo y máximo. La empresa con ánimo de lucro se financió con las importantes acciones de Albrecht (había tenido la suerte de entrar en la primera oferta pública de acciones de DEC) y pronto la empresa obtuvo un contrato para escribir una serie de libros instructivos sobre BASIC.

Albrecht y el grupo de Dymax se hicieron con una minicomputadora DEC PDP-8. Para albergar esta maravillosa máquina, trasladaron la empresa a una nueva sede en Menlo Park. Según su acuerdo con DEC, Bob recibiría una computadora y un par de terminales a cambio de escribir un libro para DEC llamado *My Computer Likes Me*, quedándose astutamente con los derechos de autor (vendería más de un cuarto de millón de copias). El equipo fue embalado en un autobús VW, y Bob revivió los días de la feria de medicina, llevando su gira de presentaciones del PDP-8 a las escuelas. Llegó más equipo, y en 1971 Dymax se convirtió en un lugar de reunión popular para jóvenes

informáticos, hackers en ciernes, aspirantes a gurús de la educación informática y descontentos tecnosociales. Bob, mientras tanto, se había mudado a un queche de catorce metros atracado en Beach Harbor, a unos treinta kilómetros al sur de la ciudad. "Nunca había navegado en mi vida. Simplemente había decidido que era hora de vivir en un barco", dijo más tarde.

Albrecht era criticado a menudo por la *multitud moderna* de Palo Alto - que creía que la tecnología es el mal - por promover las computadoras. Por este motivo su método de adoctrinar al Pueblo en el mundo de la informática se volvió sutil, un enfoque astuto de traficante de drogas: "Simplemente dale una pitada a este juego... se siente bien, ¿no?... Puedes programar esta cosa, ¿sabes?" Más tarde explicó: "Éramos encubiertos. Sin quererlo, estábamos adoptando una visión a largo plazo, alentando a cualquiera que quisiera usar computadoras, escribiendo libros que la gente pudiera aprender a programar.

“Pero en Dymax había mucha contracultura. El lugar estaba lleno de fanáticos populistas de la informática, de pelo largo, muchos de ellos en edad de ir al instituto. Bob Albrecht hacía el papel de gurú barbudo, soltando ideas y conceptos más rápido de lo que nadie podría llevarlos a cabo. Algunas de sus ideas eran brillantes, otras basura, pero todas estaban impregnadas del carisma de su personalidad, que a menudo era encantadora pero también podía ser autoritaria. Albrecht llevaba al equipo de excursión a bares de piano locales donde acababa con el micrófono en la mano, dirigiendo al grupo en festivales de canciones. Acondicionó parte de las oficinas de Dymax como una taberna griega, con luces navideñas parpadeantes, para sus clases de baile los viernes por la noche. Sin embargo, sus ideas más demoníacas implicaban la popularización de las computadoras.

Albrecht pensó que algún tipo de publicación debería registrar este movimiento, ser un pararrayos para nuevos desarrollos. Así que el grupo comenzó una publicación sumamnete sensacionalista llamada *People’s Computer Company*, en honor al grupo de rock de Janis Joplin, Big Brother and the Holding Company. En la portada del primer número, fechado en octubre de 1972, había un dibujo ondulado de un barco de aparejo cuadrado navegando hacia el atardecer, que de alguna manera simbolizaba la era dorada en la que la gente estaba entrando, y la siguiente leyenda escrita a mano:

> LAS COMPUTADORAS SE UTILIZAN PRINCIPALMENTE
>
> CONTRA EL PUEBLO EN LUGAR DE A FAVOR DEL PUEBLO.
>
> SE UTILIZAN PARA CONTROLAR AL PUEBLO EN LUGAR DE
>
> LIBERARLO.
>
> ES HORA DE CAMBIAR TODO ESO... NECESITAMOS UNA...
>
> COMPAÑÍA DE COMPUTADORAS DEL PUEBLO

El periódico tenía un estilo similar al del *Whole Earth Catalog*, solo que más improvisado y descuidado. Podía haber cuatro o cinco tipografías diferentes en una página, y a menudo los mensajes se garabateaban directamente en las páginas, demasiado urgentes para esperar al tipógrafo. Era una expresión perfecta del estilo apresurado y abarcador de Albrecht. Los lectores tenían la impresión de que no había tiempo que perder en la misión de difundir la informática entre la gente, y ciertamente no perdían el tiempo en tareas aleatorias como enderezar márgenes, diseñar

historias con prolijidad o planificar con demasiada antelación. Cada número estaba repleto de noticias sobre personas imbuidas de la religión informática, algunas de las cuales iniciaban operaciones similares en diferentes partes del país. Esta información se presentaba en misivas caprichosas, despachos desde las primeras líneas de la revolución del cómputo popular, con gran conocimiento de la informática. Hubo poca respuesta de las torres de marfil de la academia o de las instituciones de investigación de cielo azul. Los hackers como los del MIT ni siquiera pestañeaban ante el PCC, que, después de todo, imprimía listados de programas en BASIC, por el amor de Dios, no en su amado lenguaje ensamblador. Pero la nueva generación de hackers de hardware, los del tipo Lee Felsenstein que intentaban encontrar formas de conseguir más acceso a las computadoras para ellos mismos y quizás para otros, descubrieron el tabloide y escribían para ofrecer listados de programación, sugerencias sobre cómo comprar componentes de computadoras o simplemente dar ánimos. Felsenstein, de hecho, escribió una columna sobre hardware para PCC.

El éxito del periódico llevó a Dymax a escindir la operación en una empresa sin fines de lucro llamada PCC, que incluiría no solo la publicación, sino también la operación del floreciente centro de cómputo en sí, que impartía clases y ofrecía computación en las calles por cincuenta centavos la hora a cualquiera que quisiera usarlo.

PCC y Dymax estaban ubicados en un pequeño centro comercial en Menalto Avenue, en el espacio que antes ocupaba una farmacia de la esquina. El lugar estaba equipado con cabinas estilo cafetería. “Siempre que alguien quería hablar con nosotros, salíamos y comprábamos un paquete de seis cervezas y hablábamos en nuestras cabinas”, recordó Albrecht más tarde. En la zona de máquinas de al lado estaba el PDP-8, que parecía un receptor estéreo gigante con luces intermitentes en lugar de un dial de FM y una fila de interruptores en el frente. La mayoría de los muebles,- salvo algunas sillas delante de los terminales grises de estilo teletipo - consistían en grandes almohadas que la gente usaba como cojines para sentarse, camas o armas para arrojarse. Una alfombra verde descolorida cubría la zona, y contra una pared había una estantería maltratada llena de una de las mejores y más activas colecciones de ciencia ficción de bolsillo de la zona.

El aire solía estar lleno del ruido de los terminales, uno conectado al PDP-8, otro conectado a las líneas telefónicas, a través de las cuales podía acceder a un ordenador de Hewlett-Packard, que había donado tiempo libre al PCC. Lo más probable era que alguien estuviera jugando a uno de los juegos que había escrito el creciente grupo de hackers del PCC. A veces, las amas de casa traían a sus hijos, probaban las computadoras ellas mismas y se enganchaban, programando tanto que los maridos temían que las leales matriarcas estuvieran abandonando a los niños y la cocina por los placeres del BASIC. Algunos hombres de negocios intentaron programar la computadora para predecir los precios de las acciones y dedicaron cantidades infinitas de tiempo a esa quimera. Cuando se tenía un centro de computación con la puerta abierta de par en par, cualquier cosa podía pasar. Albrecht fue citado en la Saturday Review diciendo: "Nosotros queremos crear centros informáticos amigables en el vecindario, donde la gente pueda entrar como si fuera un salón de bolos o una sala de juegos y descubrir cómo divertirse con las computadoras”.

Parecía que estaba funcionando. Como muestra de lo cautivadoras que podían ser las máquinas, un reportero que estaba haciendo un reportaje sobre PCC llegó alrededor de las cinco y media un día, y los trabajadores lo sentaron frente a una terminal de teletipo que estaba ejecutando un juego llamado Star Trek. “Lo siguiente que recuerdo”, escribió el reportero en una carta a PCC, “es que alguien me tocó el hombro a las 12:30 a.m. de la mañana siguiente y me dijo que era hora de irme a casa”. Después de un par de días de pasar tiempo en PCC, el reportero concluyó: “Todavía no tengo

nada que decirle a un editor más allá de que pasé un total de veintiocho horas hasta ahora simplemente jugando juegos en estas seductoras máquinas".

Todos los miércoles por la noche, PCC tenía sus cenas compartidas. Después de una reunión de personal del PCC típicamente desorganizada (Bob, con ideas que le llegaban a la cabeza como torpedos de Spacewar, no podía seguir fácilmente una agenda), se cubrían largas mesas con manteles y, poco a poco, la sala se llenaba con un virtual quién es quién de la informática alternativa en el norte de California.

De los distinguidos visitantes que llegaban, ninguno era tan bienvenido como Ted Nelson. Nelson era el autor autoeditado del *Computer Lib*, la epopeya de la revolución informática, la biblia del sueño hacker. Fue lo suficientemente terco como para publicarlo cuando nadie más parecía pensar que era una buena idea.

Ted Nelson tenía una dolencia autodiagnosticada de estar años adelantado a su tiempo. Hijo de la actriz Celeste Holm y el director Ralph Nelson (Lilies of the Field), producto de escuelas privadas, estudiante en elegantes universidades de artes liberales, Nelson era un perfeccionista, sin duda irascible, su principal talento era el de un "innovador". Escribió un musical de rock en 1957. Trabajó para John Lilly en el proyecto Dolphin e hizo algunos trabajos cinematográficos. Pero su cabeza, como explicó más tarde, estaba "nadando en ideas" sin remedio hasta que entró en contacto con un mainframe y aprendió algo de programación.

Eso fue en 1960. Durante los catorce años siguientes, pasó de un trabajo a otro. Salía de su oficina en un puesto en una corporación de alta tecnología y veía "la increíble desolación del lugar en esos pasillos". Empezó a ver cómo la mentalidad de proceso por lotes de IBM había cegado a la gente ante las magníficas posibilidades de los ordenadores. Sus observaciones al respecto fueron universalmente desatendidas. ¿Nadie le escucharía?

Finalmente, por la ira y la desesperación, decidió escribir un "libro de contracultura informática". Ningún editor estaba interesado, ciertamente no con sus exigencias sobre el formato: un diseño similar al Whole Earth Catalog o el PCC, pero aún más libre, con páginas de gran tamaño cargadas de letra tan pequeña que apenas se podía leer, junto con anotaciones garabateadas y dibujos maníacamente amateurs. El libro estaba dividido en dos partes: una se llamaba "Computer Lib", el mundo informático según Ted Nelson; y el otro, "Dream Machines", el futuro informático según Ted Nelson. Tras desembolsar dos mil dólares de su bolsillo —"mucho para mí", diría más tarde— imprimió unos cientos de copias de lo que era un manual virtual sobre la ética hacker. Las primeras páginas gritaban con urgencia, mientras lamentaba la mala imagen general de las computadoras (culpaba de ello a las mentiras que los poderosos decían sobre ellas, mentiras que él llamaba "cibermierdas") y proclamaba en letras mayúsculas que EL PÚBLICO NO TIENE QUE TOMAR LO QUE SE LES DICE. Se declaró descaradamente un fanático de las computadoras y dijo:

> Tengo un problema. Quiero ver computadoras útiles para las personas, y cuanto antes mejor, sin que se requieran complicaciones ni servilismo humano. Cualquiera que esté de acuerdo con estos principios está de mi lado. Y cualquiera que no esté de acuerdo, no lo está.
>
> ESTE LIBRO ES POR LA LIBERTAD PERSONAL.
>
> Y CONTRA LA RESTRICCIÓN Y LA COERCIÓN...

Un cántico que puede llevar a las calles:

¡PODER INFORMÁTICO PARA EL PUEBLO!

¡ABAJO LA CIBERMIERDA!

"Las computadoras son lo que importa", decía el libro de Nelson, y aunque se vendió lentamente, se vendió, y finalmente pasó por varias reimpresiones. Mas importante, tuvo un seguimiento de culto. En el PCC, Computer Lib era una razón más para creer que pronto no sería ningún secreto que los ordenadores eran mágicos. Y a Ted Nelson lo trataban como a un rey en las cenas informales.

Pero la gente no iba a las cenas informales para ver a los magos de la revolución informática: estaban allí porque estaban interesados en los mainframes. Algunos eran hackers de hardware de mediana edad, algunos eran niños de primaria que habían sido atraídos por los ordenadores, algunos eran adolescentes de pelo largo a los que les gustaba hackear el PCC PDP-8, algunos eran educadores, algunos eran simplemente hackers. Como siempre, los planificadores como Bob Albrecht hablaban de los problemas de la informática, mientras que los hackers se concentraban en intercambiar datos técnicos o se quejaban de la predilección de Albrecht por el BASIC, al que los hackers consideraban un lenguaje "fascista" pues su estructura limitada no fomentaba el máximo acceso a la máquina y reducía el poder de un programador. No se necesitarían muchas horas antes de que los hackers se escabulleran hacia las ruidosas terminales, dejando a los activistas enfrascados en acaloradas discusiones sobre este o aquel desarrollo. Y siempre, allí estaría Bob Albrecht. Resplandeciente en el rápido progreso del gran sueño informático, estaría en el fondo de la sala, moviéndose con las iteraciones culminantes de la danza folclórica griega, hubiera música o no.

• • • • • • • •

En esa atmósfera cargada de propósito mesiánico, la gente de Community Memory se lanzó sin reservas a poner en línea su proyecto. Efrem Lipkin revisó un gran programa que sería la interfaz básica con los usuarios, y Lee se dedicó a reparar una teletipo Modelo 33 donado por la Tymshare Company. Había sido utilizado durante miles de horas y se lo había entregado a CM como basura. Debido a su fragilidad, alguien tendría que cuidarlo constantemente; A menudo se atascaba, o el regulador se ponía pegajoso, o no pulsaba el retorno de carro antes de imprimir la siguiente línea. Más adelante en el experimento, CM conseguiría un terminal Hazeltine 1500 con un CRT que era un poco más fiable, pero alguien del colectivo tenía que estar allí en caso de que hubiera un problema. La idea era que Lee desarrollara con el tiempo un nuevo tipo de terminal para mantener el proyecto en marcha, y ya estaba empezando a idear ideas para ese proyecto de hardware.

Pero eso era para más adelante. Primero tenían que llevar a CM a las calles. Después de semanas de actividad, Efrem, Lee y los demás instalaron la teletipo Modelo 33 y su carcasa de caja de cartón (que protegía contra los derrames de café y las cenizas de marihuana) en Leopold's Records. Habían dibujado carteles que instruían a la gente sobre cómo utilizar el sistema, pegatinas de papel glacé de colores brillantes con conejos psicodélicos y líneas onduladas. Se imaginaban a la gente haciendo conexiones difíciles para cosas como trabajos, lugares para vivir, viajes y trueques. Era tan simple que cualquiera podía usarlo: solo había que usar los comandos ADD o FIND. El sistema era una variación cariñosa del sueño de los hackers, y encontraron un sentimiento similar en un poema que los inspiró a otorgarle un nombre especial a la empresa matriz de Community Memory: “Loving Grace Cybernetics”. El poema era de Richard Brautigan:

Me gusta pensar (¡y cuanto antes mejor!)

en un prado cibernético

donde mamíferos y computadoras

viven juntos en armonía de programación mutua

como agua pura

tocando el cielo claro

Me gusta pensar

(¡ahora mismo, por favor!)

en un bosque cibernético

lleno de pinos y dispositivos electrónicos

donde los ciervos pasean pacíficamente

frente a las computadoras

como si fueran flores

con capullos que giran.

Me gusta pensar

(¡tiene que serlo!)

en una ecología cibernética

donde seamos libres de nuestro trabajo

y nos unamos de nuevo a la naturaleza,

devueltos a nuestros hermanos y hermanas

mamíferos,

y todos vigilados

por máquinas de gracia amorosa.

—TODOS VIGILADOS POR MÁQUINAS DE GRACIA AMOROSA

Eso no era una simple terminal en Leopold's, ¡era un instrumento de Gracia Amorosa! Era para pastorear al rebaño ignorante hacia un prado fertilizado por la benévola Ética Hacker, protegido de las ínfulas sofocantes de la burocracia. Pero algunos dentro de Community Memory tenían dudas.

Aún mayor que las persistentes dudas de Lee sobre la durabilidad del terminal era su temor de que la gente reaccionara con hostilidad ante la idea de que un ordenador invadiera el espacio sagrado de una tienda de discos de Berkeley; sus peores temores hicieron que los “pregoneros” de Community Memory que atendían el terminal se vieran obligados a proteger físicamente el hardware contra una turba de luditas hippies.

Temores infundados. Desde el primer día del experimento, la gente reaccionó con calidez ante la terminal. Tenían curiosidad por probarla y se devanaban los sesos para pensar en algo que poner en el sistema. En el Berkeley Barb una semana después de que comenzara el experimento, Lee escribió que durante los primeros cinco días de la terminal teletipo Modelo 33 en Leopold's, estuvo en uso 1.434 minutos, aceptando 151 elementos nuevos e imprimiendo 188 sesiones, el treinta y dos por ciento de las cuales representaban búsquedas exitosas. Y el nivel de violencia fue inexistente: Lee informó de “cien por ciento de sonrisas”.

Se corrió la voz y pronto la gente empezó a buscar contactos importantes. Si escribías, por ejemplo, BUSCAR CLÍNICAS DE SALUD, obtenías información sobre cualquiera de ocho, desde la Clínica de Investigación Médica de Haight-Ashbury hasta la Clínica Popular Gratuita George Jackson. Una solicitud de BAGELS (alguien que preguntaba dónde en el Área de la Bahía se podían encontrar buenos bagels al estilo neoyorquino) obtenía cuatro respuestas: tres de ellas mencionaban puntos de venta minoristas, otra de una persona llamada Michael que daba su número de teléfono y se ofrecía a mostrarle al consultante cómo hacer sus propios bagels. La gente encontraba compañeros de ajedrez, compañeros de estudio y compañeros sexuales para boas constrictoras. Pasaban consejos sobre restaurantes y álbumes de discos. Ofrecían servicios como cuidado de niños, transporte, mecanografía, lectura del tarot, plomería, pantomima y fotografía (“TIPO TRANQUILO BUSCA PERSONAS QUE SE DEDIQUEN A LA FOTOGRAFÍA/MODELO/AMBOS NO EXPLOTABLES... OM SHANTI”).

Se produjo un fenómeno extraño. A medida que avanzaba el proyecto, los usuarios comenzaron a aventurarse en aplicaciones inexploradas. Mientras la gente de la Memoria Comunitaria observaba entre las novedades del día, encontraron algunos artículos que no encajaban en ninguna categoría... Incluso las palabras clave introducidas al final del artículo eran desconcertantes. Había mensajes como: “TÚ ERES TU MEJOR AMIGA”, seguido de las palabras clave AMIGA, AMANTE, PERRA, TÚ, NOSOTRAS, NOSOTROS, GRACIAS. Había mensajes como: “ALIENÍGENA DE OTRO PLANETA NECESITA FÍSICO COMPETENTE PARA COMPLETAR LAS REPARACIONES EN LA NAVE ESPACIAL. AQUELLOS QUE NO TIENEN CONOCIMIENTOS DE INDUCCIÓN GEOMAGNÉTICA NO DEBEN SOLICITARLO”. Había mensajes como: “DIOS MÍO, ¿POR QUÉ ME HAS ABANDONADO?”. Había mensajes que incluían citas crípticas de Ginsberg, The Grateful Dead, Arlo Guthrie y Shakespeare. Y había mensajes del Doctor Benway y la misteriosa Interzona. El doctor Benway, el personaje de El almuerzo desnudo, era «un manipulador y coordinador de sistemas de símbolos, un experto en todas las fases de interrogatorio, lavado de cerebro y control». No importaba. Quienquiera que fuese este usuario demente, empezó a organizar los bits del núcleo dentro del XDS-940 en diatribas deshilachadas, comentarios superficiales de la época aderezados con visiones inefables, llamamientos a la lucha armada y predicciones funestas de gran hermandad, predicciones que se presentaban irónicamente mediante el uso de tecnología informática al estilo de 1984 de una manera radical y creativa. «Benway aquí», se anunciaba en una entrada típica, «sólo un excursionista en las arenas de esta fecunda base de datos». Benway no era el único que adoptaba personajes extraños:

como ya habían descubierto los hackers, el ordenador era una extensión ilimitada de la propia imaginación, un espejo sin prejuicios en el que se podía enmarcar cualquier tipo de autorretrato que se deseara. No importaba lo que escribieras, las únicas huellas digitales que tenía tu mensaje eran las de tu imaginación. El hecho de que a los no hackers les gustaran estas ideas indicaba que la mera presencia de ordenadores en lugares accesibles podía ser un estímulo para el cambio social, una oportunidad de ver las posibilidades que ofrecía la nueva tecnología.

Lee lo llamaría más tarde "una epifanía, una revelación. Fue como mi experiencia con el Movimiento de la Libertad de Expresión y el Parque del Pueblo. ¡Dios mío! ¡No sabía que la gente pudiera hacer esto!".

Jude Milhon desarrolló personalidades en línea, escribió poemas. "Fue muy divertido", recordaría más tarde. "Tus sueños encarnados". Un habitual de CM intercambió mensajes electrónicos con Benway, profundizando en el tema de Naked Lunch para crear una "Interzona" informática, en honor al decadente mercado de carne del alma creado por Burroughs. Al principio, los mensajes de Benway indicaban sorpresa por esta variación; luego, casi como si se diera cuenta de las posibilidades democráticas del medio, dio su bendición. "Algunos piratas nefastos han hablado de clonar el logotipo de Benway... adelante... es de dominio público", escribió.

Jude Milhon conoció a Benway. Era, como ella lo describió, "muy tímido, pero capaz de funcionar en el mundo de Community Memory".

El grupo floreció durante un año y medio, trasladando la terminal en un momento dado de Leopold's a Whole Earth Access Store, y colocando una segunda terminal en una biblioteca pública en Mission District de San Francisco. Pero las terminales seguían averiándose, y quedó claro que era esencial un equipo más confiable. Se necesitaba un sistema completamente nuevo, ya que CM sólo podía llegar hasta cierto punto con el enorme gigante XDS-940, y en cualquier caso la relación entre CM y Resource One (su fuente de financiación) se estaba rompiendo. Pero no había ningún sistema esperando entre bastidores, y Community Memory, escaso de fondos y tecnología, y ambos en rápido decline ante la energía del Pueblo, necesitaba algo pronto.

Finalmente, en 1975, un grupo agotado de idealistas de Community Memory se sentó para decidir si continuar con el proyecto. Había sido un año estimulante y agotador. El proyecto "mostró lo que se podía hacer. "Mostró el camino", afirmaría Lee más tarde. Pero Lee y los demás consideraron que era "demasiado arriesgado" continuar con el proyecto en su estado actual. Habían invertido demasiado, tanto técnica como emocionalmente, para ver cómo el proyecto se desvanecía a través de una serie de deserciones frustrantes y fallos aleatorios del sistema. El consenso fue sumergir el experimento en un estado de remisión temporal. Aun así, fue una decisión traumática. "Estábamos en desarrollo cuando se interrumpió", dijo Jude Milhon más tarde, "[Nuestra relación con] Community Memory era como Romeo con Julieta, nuestra otra mitad del alma. Luego, de repente, CHOP, se acabó. Se cortó en su florecimiento temprano".

Efrem Lipkin se fue e intentó una vez más pensar en una forma de abandonar las computadoras. Otros se involucraron en varios otros proyectos, algunos técnicos, algunos sociales. Pero ninguno, y mucho menos Lee Felsenstein, abandonó el sueño.

Capítulo 9. Todo hombre un dios

En junio de 1974, Lee Felsenstein se mudó a un apartamento de una habitación sobre un garaje en Berkeley. No tenía muchas comodidades (ni siquiera un termostato), pero solo costaba 185 dólares al mes y Lee podía colocar un banco de trabajo en la esquina y llamarlo su hogar. Prefería gastos generales bajos, portabilidad y utilidad en un solo lugar.

Felsenstein tenía un proyecto de diseño específico en mente. Una terminal de computadora construida sobre el concepto de memoria comunitaria. Lee aborrecía las terminales construidas para ser absolutamente seguras frente a usuarios descuidados, cajas negras que arrojan información y que, por lo demás, son opacas en su construcción. Creía que la gente debería tener una idea de lo que hace que la máquina funcione, y que se debería instar al usuario a interactuar en el proceso. Cualquier cosa tan flexible como las computadoras debería inspirar a la gente a participar en actividades igualmente flexibles. Lee consideraba que la computadora en sí misma era un modelo de activismo y esperaba que la proliferación de computadoras para las personas, en efecto, propagara la ética hacker en toda la sociedad, dando a la gente poder no solo sobre las máquinas sino también sobre los opresores políticos.

El padre de Lee Felsenstein le había enviado un libro de Ivan Illich titulado *Tools for Conviviality* (Herramientas para la convivencia), y las afirmaciones de Illich corroboraban las opiniones de Lee (“Para mí, los mejores maestros me dicen lo que ya sé que es correcto”, explicaría Lee más tarde). Illich profesaba que el hardware debería diseñarse no solo para la comodidad de la gente, sino con la visión a largo plazo de la eventual simbiosis entre el usuario y la herramienta. Esto inspiró a Felsenstein a concebir una herramienta que encarnara los pensamientos de Illich, Bucky Fuller, Karl Marx y Robert Heinlein. Sería una terminal para la gente. Lee la bautizó como Tom Swift Terminal, “en honor al héroe popular estadounidense que probablemente sería encontrado manipulando el equipo”. Sería Lee Felsenstein quien haría realidad el sueño hacker.

Mientras tanto, viviría de los ingresos que obtenía de contratos de ingeniería freelance. Uno de los lugares donde buscó trabajo fue Systems Concepts, la pequeña empresa que empleaba a los veteranos del MIT Stew Nelson (el mago de la telefonía y genio de la codificación) y al ex alumno de TMRC y TX-0 Peter Samson. Felsenstein desconfiaba de todo lo que tuviera que ver con el MIT; como es típico de los hackers de hardware, le ofendía lo que consideraba la excesiva pureza de esos hackers, en particular su despreocupación a la hora de difundir la tecnología entre los “perdedores”. “Cualquiera que haya estado en contacto con la inteligencia artificial es probable que sea un caso perdido”, explicaría más tarde. “Están tan alejados de la realidad que no pueden lidiar con el mundo real. Cuando empiezan a decir: ‘Bueno, básicamente todo lo que necesitas hacer es punto punto punto’, me quedo vidrioso y digo: ‘Está bien, amigo, pero esa es la parte fácil. Donde hacemos nuestro trabajo es el resto’”. Sus sospechas se confirmaron cuando conoció al diminuto pero tenaz Stew Nelson. Casi al instante, se vieron envueltos en un desacuerdo, una disputa técnica arcana que Lee más tarde denominó “la típica disputa de hackers del tipo “soy más inteligente que tú”. Stew insistía en que se podía hacer un determinado truco de hardware, mientras que Lee, cuyo estilo de ingeniería estaba formado por su paranoia de la primera infancia de que las cosas podrían no funcionar, dijo que no se arriesgaría. Sentado en la gran estructura de madera, parecida a un almacén, que albergaba Systems Concepts, Lee sintió que estos tipos no estaban tan interesados en hacer llegar la tecnología informática a la gente como en la elegante y alucinante pirotecnia

informática. Para Lee, eran jesuitas tecnológicos. No le preocupaba la alta magia que podían producir ni el exaltado panteón de magos canónicos que reverenciaban. ¿Y la gente?

Así que cuando Stew Nelson, el arquetipo del hacker del MIT, le dio a Felsenstein el equivalente a una audición, una prueba rápida de diseño para un producto de hardware, Lee no siguió el juego. No le importaba en absoluto producir el ingenio tecnológico que Stew buscaba. Lee se marchó.

Buscaría trabajo en otra parte. Pensó que podría lograrlo si ganaba ocho mil dólares al año. Debido a la recesión, había sido difícil encontrar trabajo, pero las cosas estaban mejorando. A ochenta kilómetros al sur de Berkeley, Silicon Valley empezaba a cobrar vida.

Los treinta kilómetros que separaban Palo Alto, en la península, y San José, en el extremo inferior de la bahía de San Francisco, se habían ganado el título de "Silicon Valley" por el material, hecho de arena refinada, que se utilizaba para fabricar semiconductores. Dos décadas antes, Palo Alto había sido la cuna del transistor; este avance se había aprovechado para crear la magia de los circuitos integrados (CI): diminutas redes de transistores comprimidos en chips, pequeños cuadrados cubiertos de plástico con finos conectores metálicos en la parte inferior. Parecían insectos robot sin cabeza. Y ahora, a principios de los años 70, tres ingenieros audaces que trabajaban para una empresa de Santa Clara llamada Intel habían inventado un chip llamado microprocesador: un diseño de conexiones sorprendentemente intrincado que duplicaba la compleja red de circuitos que uno encontraría en la unidad central de procesamiento (CPU) de una computadora.

Los jefes de estos ingenieros todavía estaban considerando los usos potenciales del microprocesador.

Lee Felsenstein, en cualquier caso, era reacio a arriesgarse con una tecnología completamente nueva. Su estilo de ingeniería de "caja de chatarra" impedía utilizar cualquier cosa que no fuera productos que sabía que iban a durar algún tiempo. El éxito del microchip y el rápido proceso de reducción de precios que se produjo después de que los chips se fabricaran en masa (costaba una fortuna diseñar un chip y hacer un prototipo; costaba muy poco producir un chip después de que existiera una línea de montaje para producirlos) dieron como resultado una escasez de chips en 1974, y Felsenstein tenía poca confianza en que la industria pudiera mantener estos nuevos microprocesadores en suficiente suministro para su diseño. Se imaginó a los usuarios de su terminal tratándolo de la misma manera que los hackers tratan un sistema operativo de computadora, cambiando partes y haciendo mejoras... "un sistema vivo en lugar de un sistema mecánico", explicaría más tarde. "Las herramientas son parte del proceso regenerativo". Estos usuarios necesitarían un acceso constante a las piezas. Así que, mientras esperaba que se desarrollaran ganadores claros en la carrera de los microchips, se tomó su tiempo para reflexionar sobre las lecciones de Ivan Illich, que estaba a favor del diseño de una herramienta "que mejore la capacidad de las personas para perseguir sus propios objetivos a su manera única". En los días soleados en la relajada Berkeley, Lee llevaba su mesa de dibujo al People's Park, la franja de vegetación que había ayudado a liberar en los no muy lejanos años sesenta, y hacía bocetos de esquemas, quemándose con el sol por el reflejo del papel blanco de dibujo.

Felsenstein era sólo uno de los cientos de ingenieros del Área de la Bahía que en algún momento de su vida habían dejado de lado toda pretensión de que su interés era exclusivamente profesional. Les encantaban los aspectos prácticos de los circuitos y la electrónica, y aunque muchos de ellos trabajaban de día en empresas con nombres exóticos como Zilog, Itel y National Semiconductor, volvían a casa por la noche y construían, construían proyectos fantásticos en placas serigrafiadas

con base de epoxi cargadas de líneas grabadas y filas irregulares de circuitos integrados. Soldadas en cajas de metal, las placas cumplían funciones extrañas: funciones de radio, funciones de vídeo, funciones lógicas. Menos importante que hacer que estas placas realizaran tareas era el acto de fabricar las placas, de crear un sistema que hiciera algo. Se trataba de hackear. Si había un objetivo, era construir un ordenador en la propia casa. No para que cumpliera una función específica, sino para jugar con él, para explorar. El sistema definitivo. Pero estos hackers de hardware no solían confiar su objetivo a extraños porque en 1974 la idea de que una persona normal tuviera un ordenador en su casa era patentemente absurda.

Aun así, así iban las cosas. Se podía sentir el entusiasmo en todas partes donde se reunían estos hackers de hardware. Lee participaba en debates técnicos en las comidas chinas compartidas del PCC. También asistía a las sesiones de tonterías de los sábados por la mañana en la tienda de chatarra de Mike Quinn.

La tienda de Quinn era la contraparte en el Área de la Bahía de San Francisco de Eli Heffron en Cambridge, donde los hackers de Tech Model Railroad buscaban interruptores de barras transversales y relés de paso. En la tienda, una gigantesca estructura de color gris acorazado, de estilo antiguo de la Segunda Guerra Mundial, parecida a un hangar, situada en los terrenos del aeropuerto de Oakland, se encontraba Vinnie “el Oso” Golden. En un mostrador abarrotado de cajas de resistencias e interruptores rebajados a centavos, Vinnie el Oso regateaba con los hackers de hardware a los que llamaba cariñosamente “tacaños solitarios”. Regateaban los precios de las placas de circuitos usadas, los osciloscopios excedentes del gobierno y un montón de LED (diodos emisores de luz) de relojes digitales. Moviéndose por el piso de madera desgastado de la gigantesca estructura, los hackers rebuscaban entre las filas de cajas que contenían miles de circuitos integrados, condensadores, diodos, transistores, placas de circuitos en blanco, potenciómetros, interruptores, enchufes, clips y cables. Un cartel en letras góticas decía SI NO LO ENCUENTRAS, BUSCALO y era un consejo bien tomado. Un centenar de empresas que fracasaron utilizaban Quinn’s para deshacerse de sus excedentes, y uno podía encontrarse con una unidad de control de gas gigante, una pila de cintas de ordenador usadas o incluso una unidad de cinta de ordenador usada del tamaño de un archivador. Vinnie the Bear, un gigante barbudo y panzudo, recogía las piezas que uno le ofrecía para que las observara, adivinaba los posibles límites de sus usos, se preguntaba si uno podría establecer una conexión con esta o aquella pieza y se atenía a la leyenda del cartel que tenía encima: “El precio varía según la actitud del comprador”. Se producían todo tipo de discusiones técnicas que finalmente terminaban con Vinnie the Bear murmurando vagos insultos sobre la inteligencia de los participantes, que volvían todos la semana siguiente en busca de más basura y más charlas.

Al lado de Mike Quinn’s estaba la operación de Bill Godbout, que compraba basura a una escala más masiva: normalmente chips y piezas excedentes del gobierno que se rechazaban por no cumplir los exigentes estándares requeridos para una función específica, pero que eran perfectamente aceptables para otros usos. Godbout, un piloto rudo, fornido y todavía activo en un insinuado pasado cargado de espionaje internacional e intrigas de agencias gubernamentales cuyos nombres no podía pronunciar legalmente, tomaría dichas piezas, les ponía su propia marca y las vendía, a menudo en kits de circuitos lógicos que se podían comprar por correo. Gracias a su conocimiento enciclopédico sobre lo que pedían las empresas y lo que desechaban, Godbout parecía saber todo lo que pasaba en Silicon Valley y, a medida que su operación se hacía más grande, suministraba cada vez más piezas y kits a los entusiastas hackers de hardware.

Lee llegó a conocer a Vinnie, Godbout y a docenas de personas más, pero desarrolló una relación especialmente estrecha con un hacker de hardware que se había puesto en contacto con él a través de la terminal Community Memory antes de que el experimento entrara en remisión indefinida. Era alguien a quien Lee conocía vagamente de sus días en Oxford Hall en Berkeley. Su nombre era Bob Marsh.

Marsh, un hombre pequeño, con bigote a lo Pancho Villa, pelo largo y oscuro, piel pálida y una forma de hablar tensa e irónica, había dejado un mensaje para Lee en la terminal preguntándole si quería participar en la construcción de un proyecto sobre el que Marsh había leído en un número reciente de Radio Electronics. Un artículo de un hacker de hardware llamado Don Lancaster describía cómo los lectores podían construir lo que él llamaba una “máquina de escribir de televisión”, algo que permitiría poner caracteres de un teclado tipo máquina de escribir en una pantalla de televisión, como en una elegante terminal de computadora.

Marsh había sido un fanático del hardware desde la infancia; su padre había sido operador de radio y trabajó en equipos de radioaficionados durante la escuela. Se especializó en ingeniería en Berkeley, pero se desvió y pasó la mayor parte de su tiempo jugando al billar. Abandonó los estudios, se fue a Europa, se enamoró y volvió a la escuela, pero no a la de ingeniería: eran los años sesenta y la ingeniería era extremadamente poco cool, casi de derechas. Pero sí trabajó en una tienda de alta fidelidad, vendiendo, reparando e instalando equipos de sonido, y siguió trabajando en la tienda después de graduarse con un título en biología. Imbuido de idealismo, quería ser profesor de niños pobres, pero esto no duró mucho cuando se dio cuenta de que, se lo mirara como se lo mirara, la escuela estaba regimentada: los estudiantes se sentaban en filas precisas y no podían hablar. Los años de trabajo en el mundo libre de la electrónica habían infundido a Marsh la ética hacker, y veía la escuela como un sistema ineficiente y represivo. Incluso cuando trabajaba en una escuela radical con un aula abierta, pensaba que era una farsa, una cárcel.

Así que, después de un intento fallido de dirigir una tienda de equipos de sonido (no era un muy buen hombre de negocios), volvió a la ingeniería. Un amigo llamado Gary Ingram, que trabajaba en una empresa llamada Dictran, le consiguió un trabajo en el primer voltímetro digital. Después de un par de años en eso, se interesó en la idea de las computadoras y se sorprendió al ver el artículo de Lancaster. Pensó que podría usar la TV Typewriter como terminal para conectarse a una computadora.

Comprando piezas de Mike Quinn para mejorar el equipo del kit ofrecido en la revista, trabajó durante semanas en el proyecto, tratando de mejorar el diseño aquí y allá. Nunca consiguió que funcionara al cien por cien, pero el objetivo era hacerlo, aprender sobre él. Más tarde explicó: “Era lo mismo que la radioafición. No quería gastar mi dinero para salir en antena alardeando de mi equipo. Quería construir cosas”.

Lee respondió al mensaje de Marsh en CM y se encontraron en la sede del grupo en una tienda. Lee le habló de la terminal Tom Swift, una terminal que utilizaría un televisor doméstico como pantalla de caracteres, un “bloque de construcción cibernético” que podría expandirse a casi cualquier cosa. Marsh quedó impresionado. También estaba desempleado en ese momento, y pasaba la mayor parte de su tiempo hackeando la TV Typewriter en un garaje alquilado en Fourth Street, cerca de la bahía. Marsh estaba casado y tenía un hijo; el dinero se estaba agotando. Le pidió a Lee que dividiera con él el alquiler del garaje de 175 dólares, y Lee trasladó su banco de trabajo allí.

De manera que Marsh trabajaba en su proyecto mientras urdía un plan para comprarle a Bill Godbout piezas de relojes digitales y montarlas en elegantes cajas de madera. Tenía un amigo que era un gran carpintero. Mientras tanto, Lee, presidente de la empresa unipersonal LGC Engineering Company (que debía su nombre a Loving Grace Cybernetics), trabajaba en su terminal, que era tanto una aventura filosófica como un proyecto de diseño.

A diferencia de los diseños habituales, en los que todas las piezas estarían controladas por un chip central, el proyecto de Lee tenía un complejo sistema de copias de seguridad múltiples. Tendría una "memoria" (un lugar donde se podrían almacenar caracteres) y esa memoria estaría en una "tarjeta" o placa de circuito. Otras tarjetas obtendrían los caracteres del teclado y los colocarían en la pantalla. En lugar de que un procesador dirigiera el flujo, las tarjetas estarían constantemente enviando o recibiendo: "Dame, dame, dame", dirían, en efecto, a las entradas, como el teclado. La memoria sería la encrucijada del terminal. Incluso si más adelante pusieras un microprocesador en la terminal para realizar funciones similares a las de una computadora, ese chip poderoso estaría conectado a la memoria, no ejecutando todo el proceso (tarea a la que están acostumbrados los microprocesadores). Era un diseño que consagraba el concepto de descentralización. También era la paranoia de Felsenstein saliendo a la luz. No estaba dispuesto a ceder todo el poder a un chip de mala calidad. ¿Y si esta pieza falla? ¿Y si falla aquella? Estaba diseñando como si su hermano todavía estuviera mirando por encima de su hombro, listo para soltar un sarcasmo fulminante cuando el sistema colapsara.

Pero Lee había descubierto cómo la Terminal Tom Swift podría extenderse hasta la eternidad. La imaginó como un sistema en torno al cual la gente formaría clubes, el centro de pequeñas karasses del conocimiento de la Terminal Tom Swift. Reviviría la Memoria Comunitaria, galvanizaría al mundo, sería el tema principal de conversación en las comidas compartidas de Mike Quinn y PCC, e incluso sentaría las bases para la entrada de la gente en las computadoras, lo que en última instancia derribaría el malvado régimen de IBM, que prospera gracias a la ciberbasura y la manipulación monopolística del mercado.

Pero mientras la nariz de Lee se enrojecía por el reflejo del sol en los esquemas de su notable terminal, el número de enero de 1975 de Popular Electronics ya estaba en camino a casi medio millón de suscriptores aficionados. Llevaba en la portada una imagen de una máquina que tendría un impacto tan grande en estas personas como Lee imaginó que tendría la Terminal Tom Swift. La máquina era una computadora. Y su precio era de $397.

• • • • • • • •

Fue la idea original de un extraño floridano que dirigía una empresa en Albuquerque, Nuevo México. El hombre era Ed Roberts y su empresa se llamaba MITS, abreviatura de Micro Instrumentation Telemetry Systems, aunque algunos llegarían a creer que era un acrónimo de "Man In The Street" (Hombre de la calle). Ed Roberts, un enigma incluso para sus amigos más cercanos, inspiró ese tipo de especulación. Era un gigante, de un metro noventa y dos centímetros y más de ciento ochenta kilos, y tenía una energía y una curiosidad impresionantes. Se interesaba por un tema y lo devoraba al por mayor. "Suelo consumir los estantes de las bibliotecas", explicaría más tarde. Si un día su curiosidad se despertaba por la fotografía, en una semana no solo tendría un cuarto oscuro de revelado en color completo, sino que podría hablar de trabajo con expertos. Luego se iría a estudiar apicultura o historia de Estados Unidos. El tema que más le fascinaba era la tecnología y sus usos. Su curiosidad lo convirtió, como diría uno de los primeros empleados de MITS llamado

David Bunnell, en “el aficionado por excelencia del mundo”. Y en aquellos días, ser aficionado a la electrónica digital significaba probablemente ser un hacker de hardware.

Fue el modelismo de cohetes lo que lo llevó a fundar MITS, que inicialmente producía luces intermitentes para cohetes de aficionados, de modo que los von Braun de patio trasero pudieran fotografiar las trayectorias de sus intentos de hacer agujeros en el cielo. A partir de ahí, Roberts llevó a MITS a los equipos de prueba: sensores de temperatura, generadores de barrido de audio y similares. Roberts se interesó en las cosas que utilizaban LED, así que MITS fabricó relojes digitales, tanto ensamblados como en kits, y su empresa estaba perfectamente posicionada para aprovechar los avances en la tecnología de microchips que hicieron posibles las calculadoras digitales pequeñas. También los vendió en kits y la empresa despegó, expandiéndose hasta tener casi cien empleados. Pero entonces llegaron los “peces gordos”, empresas gigantes como Texas Instruments que fabricaban sus propios microchips, y las empresas más pequeñas reaccionaron reduciendo los precios de las calculadoras a tal punto que MITS no pudo competir. “Pasamos por un período en el que nuestro costo de envío de una calculadora era de treinta y nueve dólares y se podía comprar una en una farmacia por veintinueve dólares”, recordó Roberts más tarde. Fue devastador. A mediados de 1974, la empresa de Ed Roberts tenía una deuda de trescientos sesenta y cinco mil dólares.

Pero Ed Roberts tenía algo bajo la manga. Sabía de los nuevos chips de microprocesador de Intel y sabía que era posible tomar uno y construir un ordenador a su alrededor. Un ordenador. Desde que tuvo contacto por primera vez con ellos, durante su época en la Fuerza Aérea, se había sentido asombrado por su poder y disgustado por los enrevesados pasos que uno tenía que dar para tener acceso a ellos. Alrededor de 1974, Ed Roberts hablaba a menudo con su amigo de la infancia de Florida, Eddie Currie, tanto que para reducir las facturas telefónicas habían empezado a intercambiar cintas de casete. Las cintas se convirtieron en producciones en sí mismas, con efectos de sonido, música de fondo y lecturas teatrales. Un día, Eddie Currie recibió esta cinta de Ed Roberts que no se parecía a ninguna anterior. Currie más tarde recordó a Ed, con las cadencias más emocionadas que pudo reunir, hablando de construir una computadora para las masas. Algo que eliminaría al Sacerdocio de la Computadora de una vez por todas. Usaría esta nueva tecnología de microprocesadores para ofrecer una computadora al mundo, y sería tan barata que nadie podría permitirse el lujo de no comprarla.

A continuación de la cinta, llamó a Currie. ¿La compraría si costara quinientos dólares? ¿Cuatrocientos? Lo discutió con el personal que quedaba en su empresa en crisis (el personal se había reducido a un puñado relativo) y el empleado de MITS, David Bunnell, recordaría mas tarde: “Pensábamos que estaba loco como una cabra”.

Pero cuando Ed Roberts tomó una decisión, no hubo fuerza que pudiera obligarlo a reconsiderarla. Construiría un ordenador y eso fue todo. Sabía que el chip actual de Intel, el 8008, no era lo suficientemente potente, pero cuando Intel sacó uno nuevo, el 8080, que podía soportar una gran cantidad de memoria y otro hardware, Roberts llamó a la empresa para negociar. Comprados en lotes pequeños, los chips costarían 350 dólares cada uno. Pero Roberts no estaba pensando en lotes pequeños, así que “golpeó a Intel en la cabeza” para conseguir los chips por 75 dólares cada uno.

Una vez superado ese obstáculo, hizo que su ingeniero de personal Bill Yates diseñara un “bus” de hardware, una configuración de conexiones donde los puntos del chip se conectarían a salidas (“pines”) que, en última instancia, soportarían elementos tales como memoria de cómputo y todo

tipo de dispositivos periféricos. El diseño del bus no era especialmente elegante (de hecho, más tarde los hackers se quejarían de la aleatoriedad con la que el diseñador había elegido qué punto del chip se conectaría a qué punto del bus), pero reflejaba la tenaz determinación de Ed Roberts de hacer bien su trabajo. Era un secreto a voces que se podía construir un ordenador a partir de uno de esos chips, pero nadie se había atrevido a hacerlo hasta entonces. Los peces gordos de la informática, en particular IBM, consideraban absurdo todo el concepto. ¿Qué clase de loco querría un pequeño ordenador? Incluso Intel, que fabricaba los chips, pensaba que eran más adecuados para funcionar como piezas de control de semáforos que como miniordenadores. Aun así, Roberts y Yates trabajaron en el diseño de la máquina, a la que Bunnell instó a Roberts a llamar "Pequeño Hermano", en un ataque orwelliano a los peces gordos. Roberts confiaba en que la gente compraría el ordenador una vez que lo ofreciera en forma de kit. Tal vez incluso unos pocos cientos de compradores en el primer año.

Mientras Ed Roberts trabajaba en su prototipo, un editor de revistas de Nueva York, bajito y calvo, pensaba en la misma línea que Roberts. Les Solomon era un vagabundo de una historia de Bernard Malamud, un ex ingeniero nacido en Brooklyn, gracioso y con un sentido del humor de muerte. Este tipo de aspecto anodino se jactaba de un pasado como mercenario sionista que luchaba junto a Menachem Begin en Palestina. También hablaba de extraños viajes que lo llevaron a los pies de brujos indios sudamericanos, o médicos brujos, con quienes compartía drogas rituales e ingería datos previamente protegidos sobre el significado de la existencia. En 1974, buscaba a alguien que hubiera diseñado un kit de computadora para que los lectores locos por la electrónica de la revista para la que trabajaba, Popular Electronics, estuvieran a la vanguardia de la tecnología y tuvieran muchos proyectos extraños para desarrollar. Más tarde, Solomon intentaría hacer caso omiso de cualquier motivo cósmico. "Solo hay dos tipos de gratificación que un ser humano puede poseer", diría, "ego y billetera. Eso es todo, nena. Si tienes eso, estás en el negocio. Mi trabajo era conseguir artículos. Había otra revista [Radio Electronics], que también hacía cosas digitales. Sacaron un kit de computadora basado en el Intel 8008. Sabía que el 8080 podía superarlo. Hablé con Ed Roberts, que había publicado cosas sobre sus calculadoras en nuestra revista, sobre su computadora, y me di cuenta de que sería un gran proyecto en la revista. Con suerte, obtendría un aumento".

Pero Solomon sabía que este no era solo un proyecto más y, de hecho, había muchos factores aquí más allá del ego y la billetera. Esto era una computadora. Más tarde, cuando lo convencieron, Les Solomon hablaría en susurros sobre el proyecto que estaba a punto de presentar a sus lectores: "La computadora es una caja mágica. Es una herramienta. Es una forma de arte. Es el arte marcial definitivo... No hay tonterías allí. Sin la verdad, la computadora no funcionará. No se puede mentir a un ordenador, maldita sea, el bit está ahí o no está ahí". Sabía del acto de creación que es una consecuencia natural de trabajar con el ordenador con la pasión obsesiva de un hacker. "Es donde cada hombre puede ser un dios", diría Les Solomon.

Así que estaba ansioso por ver la máquina de Ed Roberts. Ed Roberts le envió el único prototipo por transporte aéreo, y se perdió en el camino. El único prototipo. Así que Solomon tuvo que mirar los esquemas, creyendo en la palabra de Roberts de que la cosa funcionaba. Él creyó. Una noche, le preguntó a su hija con ligereza cuál podría ser un buen nombre para esta máquina, y ella mencionó que en el programa de televisión Star Trek esa noche, la buena nave Enterprise estaba despegando hacia la estrella llamada Altair. Así fue que el ordenador de Ed Roberts se llamó Altair.

Roberts y su ayudante de diseño Bill Yates escribieron un artículo describiéndolo. En enero de 1975, Solomon publicó el artículo, con la dirección de MITS y la oferta de vender un kit básico por

397 dólares. En la portada de ese número había una imagen falsificada del Altair 8800, que era una caja azul del tamaño de la mitad de un acondicionador de aire, con un atractivo panel frontal cargado de pequeños interruptores y dos filas de LED rojos. (Este panel frontal se cambiaría por uno aún más elegante orlado por una tira cromada con el logotipo de MITS y la leyenda “Altair 8800” en la fuente abigarrada que se identifica con los lectores de computadora.)

Quienes leyeran el artículo descubrirían que sólo había 256 bytes (un “byte” es una unidad de ocho bits) de memoria dentro de la máquina, que venía sin dispositivos de entrada o salida; en otras palabras, era una computadora sin una forma incorporada de obtener información hacia o desde el mundo además de esos interruptores en el frente, por los cuales se podía alimentar laboriosamente la información directamente a las ubicaciones de memoria. La única forma en que podía hablar con usted era mediante las luces intermitentes en el frente. Para todos los efectos prácticos, era sorda, muda y ciega. Pero, como una persona totalmente paralizada cuyo cerebro estaba vivo, su caparazón no comunicativo oscurecía el hecho de que un cerebro de computadora estaba vivo y funcionando en su interior. Era una computadora, y lo que los hackers podrían hacer con ella estaría limitado sólo por su propia imaginación.

Roberts esperaba que llegaran unos cuatrocientos pedidos mientras MITS perfeccionaba su línea de montaje hasta el punto en que estuviera lista para procesar kits confiables para los aficionados dedicados. Sabía que estaba apostando su empresa por el Altair. En su lluvia de ideas original había hablado de difundir la informática a las masas, permitiendo que la gente interactuara directamente con las computadoras, un acto que difundiría la ética hacker por todo el país. Ese tipo de discurso, admitió más tarde, tenía un elemento de promoción. Quería salvar su empresa. Antes de que saliera el artículo, rara vez dormía, preocupado por una posible quiebra, una jubilación forzada.

El día que la revista llegó a los suscriptores estaba claro que no habría ningún desastre. Los teléfonos comenzaron a sonar y no dejaron de sonar. Y el correo traía pedidos, cada uno de los cuales incluía cheques o giros postales por cientos de dólares en equipos de MITS, no solo computadoras, sino las placas complementarias que harían que las computadoras fueran más útiles. Placas que ni siquiera habían sido diseñadas todavía. En una tarde, MITS recibió pedidos de cuatrocientas máquinas, la respuesta total que Ed Roberts se había atrevido a esperar. Y habría cientos más, cientos de personas en todo Estados Unidos que tenían deseos ardientes de construir sus propios ordenadores. En tres semanas, el estatus de MITS en su banco pasó de un valor negativo a más de 250.000 dólares.

¿Cómo describió Les Solomon el fenómeno? "La única palabra que me vino a la mente fue 'magia'. Compras el Altair, tienes que construirlo, luego tienes que construir otras cosas para conectarlo para que funcione. Eres una persona extraña. Porque solo la gente extraña se sienta en cocinas y sótanos y lugares a todas horas de la noche, soldando cosas a placas para hacer que las máquinas hagan flickety-flock. El peor horror, lo más horroroso es que aquí hay una empresa en Albuquerque, Nuevo México, de la que nadie ha oído hablar nunca. Y construyen una máquina que es un ordenador. Y una revista que publica este artículo y lo pone en la portada dice. “Ahora puedes construir tu propio ordenador por cuatrocientos dólares. Todo lo que tienes que hacer es enviar un cheque al MITS en Albuquerque y te enviarán una caja de piezas”. La mayoría de la gente no enviaría quince centavos a una empresa por un dial de linterna, ¿verdad? Unas dos mil personas, sin ser vistas, enviaron cheques, giros postales, de trescientos, cuatrocientos, quinientos dólares cada uno, a una empresa desconocida en una ciudad relativamente desconocida, en un estado técnicamente desconocido. Esas personas eran diferentes. Eran aventureros en una nueva tierra.

Eran las mismas personas que se fueron al Oeste en los primeros días de América. Los raros que decidieron que iban a California, a Oregón, o Dios sabe dónde".

Eran hackers. Sentían tanta curiosidad por los sistemas como los hackers del MIT, pero, al carecer de acceso diario a las PDP-6, tuvieron que construir sus propios sistemas. Lo que resultaría de estos sistemas no era tan importante como el acto de comprender, explorar y cambiar los sistemas mismos: el acto de creación, el ejercicio benévolo del poder en el mundo lógico e inequívoco de las computadoras, donde la verdad, la apertura y la democracia existían en una forma más pura que la que se podía encontrar en cualquier otro lugar.

Ed Roberts habló más tarde del poder: "Cuando hablas de riqueza, lo que en realidad estás diciendo es: '¿A cuántas personas controlas?' Si te diera un ejército de diez mil personas, ¿podrías construir una pirámide? Una computadora le da a la persona promedio, un estudiante de primer año de secundaria, el poder de hacer cosas en una semana que todos los matemáticos que vivieron hasta hace treinta años no podían hacer".

Un ejemplo típico de las personas que se sintieron galvanizadas por el artículo de Altair fue un contratista de construcción de Berkeley de treinta años, con cabello largo y rubio y ojos verdes brillantes llamado Steve Dompier. Un año antes de que saliera el artículo de Popular Electronics, había conducido por la empinada y sinuosa carretera que se encuentra sobre Berkeley y que conduce al Lawrence Hall of Science, una enorme y siniestra estructura de hormigón con forma de búnker que fue el escenario de la película The Forbin Project, sobre dos ordenadores inteligentes que colaboran para apoderarse del mundo. Este museo y centro educativo se financió con una subvención para apoyar la alfabetización en ciencias y, a principios de los años 70, su programa de educación informática lo dirigía uno de los presentadores originales de Bob Albrecht en los programas de medicina, Bob Kahn. Tenía una gran computadora HP de tiempo compartido conectada a docenas de terminales de teletipo de color gris metalizado y, cuando Steve Dompier visitó la sala por primera vez, hizo cola para comprar un boleto de cincuenta centavos por una hora de tiempo de computadora, como si estuviera comprando un paseo en una montaña rusa. Miró las exhibiciones mientras esperaba su turno en una terminal y, cuando llegó el momento, entró en una sala con treinta teletipos ruidosos. Se sentía como estar dentro de una mezcladora de cemento. Encendió la terminal y, con violenta confianza, la impresora de líneas escribió las palabras HOLA. ¿CÓMO TE LLAMAS? Escribió STEVE. La impresora de líneas escribió HOLA STEVE ¿QUÉ QUIERES HACER? y Steve Dompier se quedó atónito.

Más tarde lo describió así: "Era la máquina mágica que tenía inteligencia. Por supuesto, no entendía cómo funcionaba, pero en la cara de todos se veía lo mismo durante los primeros cuatro o cinco meses, hasta que comprendían que en realidad no era inteligente. Esa es la parte adictiva, esa primera magia en la que esta máquina te responde y hace cálculos matemáticos increíblemente rápidos". Para Steve Dompier, la adicción continuó. Jugaba juegos en el sistema, como Star Trek, o mantenía un diálogo con una versión del programa ELIZA de Joseph Weizenbaum. Consiguió un libro de programación BASIC y trabajó en la creación de pequeñas rutinas. Leyó Computer Lib y se politizó tecnológicamente. Compró un teletipo para su casa para poder acceder al ordenador de Lawrence Hall por teléfono, donde jugaba al nuevo juego espacial Trek '73 durante horas y horas. Y entonces oyó hablar del Altair.

Inmediatamente llamó a Albuquerque para pedir su catálogo, y cuando lo recibió, todo parecía estar bien: el kit informático, las unidades de disco opcionales, los módulos de memoria, los módulos de

reloj. Así que mandó a buscar todo. Cuatro mil dólares. Su excusa para sí mismo fue que usaría su nuevo sistema informático para catalogar todas sus revistas de Popular Science; si se preguntaba dónde estaba ese artículo sobre, por ejemplo, los tubos de calor, tecleaba TUBOS DE CALOR en el ordenador y decía: NÚMERO 4, PÁGINA 76, STEVE! Diez años y muchos ordenadores después, todavía no habría podido hacer esa tarea. Porque en realidad quería un ordenador para hackear, no para hacer un índice estúpido.

MITS le respondió diciendo que había enviado demasiado dinero; la mitad del equipo que había pedido estaba sólo en vagas etapas de planificación. La otra mitad del equipo que había pedido tampoco existía, pero MITS estaba trabajando en esos productos. Así que Steve Dompier esperó.

Esperó aquel enero, esperó aquel febrero y, a principios de marzo, la espera se había vuelto tan insoportable que se dirigió al aeropuerto, se subió a un avión, voló a Albuquerque, alquiló un coche y, armado sólo con el nombre de la calle, empezó a recorrer Albuquerque en busca de aquella empresa informática. Había estado en varias empresas de Silicon Valley, así que supuso que sabía qué buscar... un edificio largo y modernista de una sola planta sobre un gran césped verde, con aspersores zumbando y un cartel en la entrada con la palabra «MITS» grabado en madera rústica. Pero el barrio donde parecía estar la dirección no se parecía en nada a eso. Era una zona industrial destartalada. Después de conducir de un lado a otro unas cuantas veces, vio un pequeño cartel: «MITS», en la esquina de una ventana de un pequeño centro comercial, entre un salón de masajes y una lavandería de autoservicio. Si hubiera mirado en el aparcamiento cercano, habría visto un remolque en el que un hacker había estado viviendo durante las últimas tres semanas mientras esperaba que su máquina estuviera lista para ser entregada.

Dompier entró y vio que la sede de MITS eran dos pequeñas oficinas con una secretaria tratando de lidiar con un teléfono que sonaba tan pronto como se colgaba el auricular. Iba asegurando a un interlocutor tras otro que sí, que un día llegaría la computadora. Dompier conoció a Ed Roberts, que se estaba tomando todo esto con buen humor. Roberts contó una historia dorada sobre el futuro de las computadoras, cómo MITS iba a ser más grande que IBM, y luego entraron en la habitación trasera, apilada hasta el techo con piezas, donde un ingeniero sostenía un panel frontal en una mano y un puñado de LED en la otra. Y eso era todo lo que había del Altair hasta ahora.

El sistema de entrega de kits de MITS no se ajustaba del todo a las regulaciones postales de los Estados Unidos, que desaprobaban aceptar dinero por correo por artículos que no existían excepto en imágenes en las portadas de las revistas. Pero la oficina de correos no recibió muchas quejas. Cuando el amigo de Ed Roberts, Eddie Currie, se unió a la empresa para ayudar en la crisis, descubrió que su experiencia con algunos clientes de MITS en Chicago era típica: un tipo en particular se quejó de haber enviado más de mil dólares hacía más de un año antes, sin respuesta. “¡Me están estafando, ni siquiera me ofrecen devolverme el dinero!”, gritó. Currie dijo: “Está bien, deme su nombre, haré que el departamento de contabilidad se ocupe del problema. Le enviaré un cheque inmediatamente, con intereses”. El hombre rápidamente se volvió humilde. “Oh, no. No quiero eso”. Quería su equipo. “Esa era la mentalidad”, recordó Currie más tarde. “Era increíble la cantidad de gente que lo quería”.

Ed Roberts estaba en un momento de euforia, demasiado ocupado tratando de hacer las cosas como para preocuparse por lo atrasada que estaba su compañía con los pedidos. Tenía pedidos de más de un millón de dólares y planes que eran mucho más grandes que eso. Cada día, parecía, aparecían cosas nuevas que dejaban aún más claro que la revolución informática había ocurrido allí mismo.

Incluso Ted Nelson, autor de Computer Lib, llamó con su bendición. Bob Albrecht también llamó y dijo que escribiría un libro sobre juegos en el Altair, si Roberts le enviaba un modelo funcional para que lo revisara para PCC.

Finalmente, MITS logró enviar algunos kits. Steve Dompier había dejado la oficina sólo después de que Roberts le hubiera dado una bolsa de plástico con piezas con las que podía empezar a trabajar, y durante los dos meses siguientes llegarían más piezas por UPS, y finalmente Dompier tenía suficientes piezas para montar un Altair con un número de serie cuatro. El número tres fue para el tipo del aparcamiento que trabajaría con un sistema de soldadura alimentado por batería. Cada vez que tenía un problema saltaba del remolque y molestaba a un ingeniero de MITS hasta que entendía el problema. Un prototipo ensamblado incluso antes fue a PCC, que tenía la fantástica ventaja de obtener un modelo ya construido.

No era fácil montar un Altair. Eddie Currie lo reconoció más tarde cuando dijo: "Una de las cosas buenas del kit [desde el punto de vista de MITS] era que no tenías que probar las piezas que enviabas, no tenías que probar las subunidades, no tenías que probar las unidades terminadas. Simplemente ponías todo el material en sobres y lo enviabas. Le dejaba al pobre cliente la tarea de averiguar cómo montar todas esas bolsas de basura". (En realidad, explicaría Ed Roberts, habría sido más barato ensamblar las cosas en la fábrica, ya que los aficionados frustrados a menudo enviaban sus máquinas semiterminadas a MITS, que terminaría la tarea con pérdidas).

Fue una educación en sí misma, un curso de lógica digital y habilidades de soldadura e innovación. Pero se podía hacer. El problema era que cuando terminabas, lo que tenías era una caja de luces parpadeantes con solo 256 bytes de memoria. Podías introducir un programa solo pulsando números octales en el ordenador mediante esos pequeños interruptores que destrozaban los dedos, y podías ver la respuesta a tu problema solo interpretando el parpadeo de las luces LED, que también estaban dispuestas en octal. ¡Diablos, qué importaba! Era un comienzo. Era una computadora.

En la People’s Computer Company, el anuncio del Altair 8800 fue motivo de celebración. Todo el mundo sabía de los intentos de poner en marcha un sistema en lugar del chip Intel 8008, menos potente; La publicación hermana no oficial de PCC era el Micro-8 Newsletter, un documento organizado de forma bizantina con tipos microscópicos publicado por un profesor y un fanático del 8008 en Lompoc, California. Pero se hablaba del Altair, con su precio increíblemente bajo y su chip 8080, como si fuera la Segunda Venida.

El primer número de PCC en 1975 dedicó una página a la nueva máquina, instando a los lectores a hacerse con el artículo de Popular Electronics e incluyendo un apéndice escrito a mano por Bob Albrecht: “Pondremos nuestros chips en el chip. Si estás ensamblando un ordenador doméstico, un ordenador escolar, un ordenador de memoria comunitaria... un ordenador para jugar y divertirse... usando un Intel 8008 o un Intel 8080, ¡escribe una carta al Dragón de PCC!”.

Lee Felsenstein, que estaba haciendo reseñas de hardware para PCC, estaba ansioso por ver la máquina. El mayor avance anterior había sido la TV Typewriter, en la que su compañero de garaje Bob Marsh había estado trabajando, y Lee había estado en contacto con su diseñador, Don Lancaster. El diseño parecía tener el defecto fatal de borrar todo lo que había antes al final de cada página de texto (un esquema de “derviche giratorio” que borraba lo anterior cuando se actualizaba la pantalla con una nueva salida) y Lee había estado pensando en diseñar una placa para solucionarlo. Pero cuando salió el Altair, todo estaba perdido. Felsenstein y Marsh leyeron el artículo de Popular Electronics y al instante se dieron cuenta de que el modelo que aparecía en la

revista era una maqueta, y que incluso cuando el Altair real estuviera listo, sería una caja con luces intermitentes. ¡No había nada dentro! Era simplemente una extensión lógica de lo que todos sabían y de lo que nadie se había atrevido a sacar provecho.

Esto no molestó a Lee en lo más mínimo; sabía que la importancia del Altair no era como un avance tecnológico, ni siquiera como un producto útil. El valor estaría en el precio y en la promesa, y ambos incitarían a la gente a pedir kits y construir sus propias computadoras. Lee, que no tenía ningún respeto por las universidades elitistas de torre de marfil como el MIT, estaba exultante con la inauguración de la primera universidad con una especialidad en hacking de hardware: la Universidad de Altair. El título vendría después de completar cursos en Soldadura, lógica digital, improvisación técnica, depuración y saber a quién pedir ayuda. Entonces estaría listo para una matriculación de por vida hacia un doctorado en “Hacer que la Cosa Haga Algo”.

Cuando Altair envió una de las primeras computadoras ensambladas a PCC, Bob Albrecht se la prestó a Lee durante una semana. La llevó a la casa de Efrem Lipkin y la dejaron allí, tratándola como una curiosidad, una pieza de escultura. Lee desarmó la cosa y comenzó a soñar con cosas que poner en ella para hacer un sistema a partir de la máquina. En su reseña de la máquina en PCC, que se ejecutó con una imagen de un rayo cayendo sobre un pequeño pueblo, escribió: "El Altair 8800 tiene dos cosas (al menos) a su favor: está aquí y funciona. Estos hechos por sí solos garantizarán que sea LA computadora amateur durante al menos el próximo año...".

PCC dedicó páginas a la máquina, que era el centro de la revolución ahora inminente. Pero por más entusiasta que estuviera Bob Albrecht con el Altair, seguía pensando que lo más importante que su operación tenía para ofrecer era la magia inicial de la computación en sí, no la locura de los hackers de hardware que se apresuraban a encargar Altairs. Había mucha gente de hardware en PCC, pero cuando uno de ellos, Fred Moore, un idealista con algunas ideas muy políticas sobre las computadoras, le preguntó a Albrecht si podía dar una clase en PCC sobre hardware de computadoras, Albrecht se negó.

Se trataba de un conflicto clásico entre hacker y planificador. Albrecht, el planificador, quería que la magia se extendiera por todas partes, y consideraba que el intenso fanatismo del hacking de alto nivel era secundario. Los hackers de hardware querían llegar hasta el fondo de las máquinas, tan profundamente que llegaran al punto en que el mundo estuviera en su forma más pura, donde “el bit está ahí o no está ahí”, como dijo Lee Solomon. Un mundo en el que la política y las causas sociales fueran irrelevantes.

Resulta irónico que fuera Fred Moore quien quisiera liderar ese descenso hacia los misterios del hardware, porque, a su manera, Moore era mucho más un planificador que un hacker.

El interés de Fred Moore por los ordenadores no se debía sólo al placer que proporcionaban a los programadores devotos, sino también a su capacidad para unir a la gente. Fred era un activista vagabundo, un estudioso de la no violencia que creía que la mayoría de los problemas podrían resolverse si la gente pudiera reunirse, comunicarse y compartir soluciones. A veces, al servicio de estas creencias, Feed Moore hacía cosas muy extrañas.

Uno de sus momentos más notables había tenido lugar cuatro años antes, en 1971, durante la fiesta de despedida del Whole Earth Catalog. El editor Stewart Brand había causado un gran revuelo en esta fiesta de despedida del Catálogo al anunciar que iba a regalar veinte mil dólares: los mil quinientos asistentes a la fiesta debían decidir a quién dárselos. El anuncio se hizo a las 22.30 horas y durante las diez horas siguientes la fiesta pasó de asamblea municipal a conferencia

parlamentaria, a debate, a pelea, a circo y a sesión de quejas. La multitud estaba menguando: alrededor de las 3.00 horas se lanzó el I Ching, con resultados no concluyentes. Fue entonces cuando habló Fred Moore. Descrito más tarde por un periodista como “un hombre joven con cabello ondulado y barba y una expresión intensa y seria”, Moore estaba molesto porque se estaba etiquetando al dinero como un salvador y se estaba comprando a la gente. Pensó que todo el asunto se estaba volviendo deprimente. Anunció a la multitud que más importante que el dinero era el evento que estaba ocurriendo en ese momento. Señaló que un poeta había pedido dinero para publicar un libro de poemas y alguien había dicho: “Sabemos dónde puedes conseguir papel”, y otra persona había sugerido una imprenta barata... y Fred pensó que tal vez la gente no necesitaba dinero para conseguir lo que quería, sólo a sí misma. Para ilustrar el punto, Fred comenzó a prender fuego a billetes de dólar. Entonces la gente decidió votar si se molestaban en gastar el dinero; Moore se opuso a la votación, ya que votar, en su opinión, era una forma de dividir a la gente entre sí. Su oposición al concepto de votar confundió tanto el asunto que la encuesta entre la audiencia no funcionó. Luego, después de mucho más hablar, Moore comenzó a circular una petición que decía, en parte, "Creemos que la unión de la gente aquí esta noche es más importante que el dinero, un recurso mayor", e instó a la gente a firmar sus nombres en un trozo de papel para mantenerse en contacto a través de una red pragmática. Finalmente, mucho después del amanecer, cuando quedaban alrededor de veinte personas, dijeron al diablo con todo y le dieron el dinero a Fred Moore. Para citar el relato de un periodista de Rolling Stone, "Moore pareció obtener el dinero por defecto, por persistencia... Moore deambuló por ahí durante un tiempo, desconcertado y atemorizado, tratando de conseguir jinetes que lo acompañaran de regreso a Palo Alto y preguntándose en voz alta si debería depositar el dinero en una cuenta bancaria... Luego se dio cuenta de que no tenía cuenta bancaria”.

Fred Moore nunca puso el dinero en un banco (“Hacen la guerra”, dijo), pero finalmente distribuyó miles de dólares a grupos dignos. Pero la experiencia le mostró dos cosas. Una, que sabía: el dinero era malo. La otra, era el poder del Pueblo Unido, cómo podían hacer cosas sin dinero, simplemente uniéndose y usando sus recursos naturales. Por eso Fred Moore se entusiasmó tanto con las computadoras.

Moore había estado involucrado con las computadoras durante algunos años, desde que entró al centro de computación en el Centro Médico de Stanford en 1970. Viajaba en ese entonces en un autobús Volkswagen con su hija pequeña, y a veces la dejaba en el autobús mientras jugaba con la computadora. Una vez se enredó tanto con la máquina que un policía vino al centro de computación y preguntó si alguien sabía algo sobre la niña que habían dejado en el estacionamiento...

Veía la computadora como un facilitador increíble, una manera para que las personas obtuvieran el control de su entorno. Podía verlo en los niños a los que enseñaba juegos, en las clases en PCC. Los niños simplemente jugaban y se divertían. Fred impartía unas trece de estas clases por semana y pensaba mucho en cómo las computadoras podrían mantener juntas a personas alternativas en grandes bases de datos. Y luego se anunció el Altair, y pensó que las personas deberían reunirse y enseñarse mutuamente cómo usarlo. No sabía mucho de hardware, tenía poca idea de cómo construir el aparato, pero pensó que la gente de la clase se ayudaría entre sí y conseguirían resultados.

A Bob Albrecht no le gustó la idea, así que no hubo clase de hardware.

Fred Moore se puso a hablar de esto con otro frustrado parásito de la órbita del PCC, Gordon French, el ingeniero consultor que había construido —«hecho en casa», como lo llamaban los hackers de hardware— un ordenador que más o menos funcionaba, centrado en el chip Intel 8008. Llamó a su sistema Chicken Hawk. A Gordon French le gustaba construir ordenadores de la misma manera que a la gente le gusta sacar motores de los automóviles y reconstruirlos. Era un tipo desgarbado con una sonrisa amplia y torcida y un pelo largo y prematuramente gris. Le encantaba hablar de ordenadores y, a veces, cuando Gordon French se ponía a hablar del tema, parecía que se abría un grifo que no se detenía hasta que un escuadrón de fontaneros con grandes llaves inglesas y batas de goma venía a cortar el flujo. El anhelo de conocer gente con gustos similares lo llevó a PCC, pero French no tuvo éxito en su solicitud para ser parte de la junta directiva de PCC. También estaba descontento con que las comidas compartidas de los miércoles parecieran estar desapareciendo. El Altair estaba en venta, la gente se estaba volviendo loca, era hora de reunirse y no había manera de hacerlo. Entonces French y Moore decidieron comenzar un grupo de personas interesadas en construir computadoras. Su propio grupo de hardware, y estaría lleno de buenas conversaciones sobre computadoras, técnicas electrónicas compartidas y tal vez una demostración o dos de las últimas cosas que se podían comprar. Solo un grupo de hackers de hardware viendo qué podría resultar de un encuentro algo más que aleatorio.

Entonces, en los carteles publicitarios cruciales de la zona (en PCC, en Lawrence Hall, en algunas escuelas y corporaciones de alta tecnología), Fred Moore colocó un cartel que decía:

> GRUPO DE USUARIOS DE COMPUTADORAS AFICIONADOS CLUB DE COMPUTADORAS CASERAS... lo que sea
>
> ¿Está construyendo su propia computadora? ¿Terminal? ¿Televisor? ¿Máquina de escribir? ¿Dispositivo de E/S? ¿O alguna otra caja de magia negra digital?
>
> ¿O estás comprando tiempo en un servicio de tiempo compartido?
>
> Si es así, tal vez te gustaría asistir a una reunión de personas con intereses similares. Intercambiar información, intercambiar ideas, ayudar a trabajar en un proyecto, lo que sea...

La reunión se convocó para el 5 de marzo de 1975 en la dirección de Gordon en Menlo Park. Fred Moore y Gordon French acababan de preparar el terreno para el último florecimiento del sueño hacker.

Capítulo 10. El Club de Computadoras Caseras

El cinco de marzo fue una noche lluviosa en Silicon Valley. Los treinta y dos participantes en la primera reunión del grupo aún sin nombre podían escuchar la lluvia mientras estaban sentados en el duro piso de cemento del garaje para dos autos de Gordon French.

Algunas de las personas en la reunión se conocían entre sí; otras habían entrado en contacto al azar a través del volante que Fred Moore había publicado. Lee Felsenstein y Bob Marsh habían conducido desde Berkeley en la camioneta destartalada de Lee. Bob Albrecht había venido para darle al grupo su bendición y para mostrar el Altair 8800 que MITS había prestado a PCC. Tom Pittman, un ingeniero independiente que había construido una improbable computadora casera alrededor del chip 4004 de Intel, había conocido a Fred Moore en una conferencia de computadoras el mes anterior y estaba ansioso por conocer a otras personas con intereses similares. Steve Dompier, que todavía esperaba el resto de sus piezas de Altair, había visto el anuncio publicado en Lawrence Hall. Marty Spergel tenía un pequeño negocio de venta de componentes electrónicos y pensó que sería una buena idea charlar con algunos ingenieros sobre chips. Un ingeniero de Hewlett-Packard llamado Alan Baum había oído hablar de la reunión y se preguntaba si la charla sería sobre los nuevos ordenadores de bajo coste; llevó consigo a un amigo que conocía desde el instituto, un compañero de trabajo de HP llamado Stephen Wozniak.

Casi todas las personas del garaje eran apasionadas del hardware, con la posible excepción de Fred Moore, que imaginaba una especie de grupo social en el que la gente se "esforzaría" por aprender sobre hardware. No se dio cuenta de que se trataba, como diría más tarde Gordon French, "de la mejor colección de ingenieros y técnicos que se podía reunir bajo un mismo techo". Se trataba de gente muy interesada en tener ordenadores en sus casas para estudiar, jugar con ellos, crear... y el hecho de que tuvieran que construir los ordenadores no era un impedimento. La presentación del Altair les había dicho que su sueño era posible, y ver a otros con el mismo objetivo era una emoción en sí misma. Y en la parte delantera del abarrotado taller del garaje de Gordon French (nunca habría cabido un coche allí, y mucho menos dos), allí estaba, un Altair. Bob Albrecht lo encendió y las luces destellaron y todos supieron que dentro de ese implacable panel frontal había pequeños bits binarios hirviendo. LDA-ing y JMP-ing y ADD-ing.

Fred Moore había instalado una mesa en la parte delantera y tomaba notas, mientras que Gordon French, que estaba indescriptiblemente orgulloso de su propia configuración casera del 8008, moderaba. Todos se presentaron y resultó que seis de los treinta y dos habían construido su propio sistema informático de algún tipo, mientras que varios otros habían pedido Altairs. De inmediato, hubo un debate sobre los méritos relativos de los chips, en particular el 8008. De hecho, hubo un sinfín de temas para debatir: hexadecimal (números de base 16) frente a octal (base 8); códigos operativos para el 8080; almacenamiento en cinta de papel versus casete versus listados en papel y lápiz... Discutieron lo que querían en un club, y las palabras que la gente usó más fueron "cooperación" y "compartir". Se habló de lo que la gente podría hacer con las computadoras en el hogar, y algunos sugirieron juegos, control de los servicios domésticos, edición de textos, educación. Lee mencionó Community Memory. Albrecht distribuyó el último número de PCC. Y Steve Dompier contó sobre su peregrinación a Albuquerque, cómo MITS estaba tratando de completar cuatro mil pedidos, y cómo estaban tan ocupados tratando de enviar los equipos básicos que no podían siquiera pensar en enviar el material adicional que permitiría a la máquina hacer más que encender sus luces.

Fred Moore estaba muy entusiasmado con la energía generada por la reunión. Le parecía que había puesto algo en movimiento. No se dio cuenta en ese momento de que la fuente del calor intelectual no era una contemplación de tipo planificador de los cambios sociales posibles por la computación masiva, sino la fascinación candente de los hackers por la tecnología. Alentado por la disposición que todos parecían tener para trabajar juntos, Moore sugirió que el grupo se reuniera cada quince días. Como para simbolizar el concepto de libre intercambio que encarnaría el grupo, Marty Spergel, el proveedor de piezas eléctricas que sería conocido como “el hombre de la chatarra” dentro del grupo, levantó un chip Intel 8008 justo cuando todos se marchaban. “¿Quién quiere esto?”, preguntó, y cuando se levantó la primera mano, arrojó el chip, el trozo de tecnología del tamaño de una uña que podría proporcionar un buen porcentaje de la potencia multimillonaria del TX-0.

Más de cuarenta personas acudieron a la segunda reunión, que se celebró en el laboratorio de inteligencia artificial de Stanford en las colinas, hogar de los hackers al estilo Tolkien del tío John McCarthy. Gran parte de la reunión se dedicó a una discusión sobre cómo debería llamarse el grupo. Entre las sugerencias se encontraban el Club de Computación Infinitesimal, Midget Brains, el Club de Computación Steam Beer, el Club de Computación Popular, Eight-Bit Byte Bangers, el Grupo de Experimentadores Informáticos del Área de la Bahía y el Club de Computación Amateur de Estados Unidos. El club se decidió por Bay Area Amateur Computer Users Group—Homebrew Computer Club. Las tres últimas palabras se convirtieron en la denominación de facto. Con un auténtico espíritu hacker, el club no exigía membresía, no pedía cuotas mínimas (aunque la sugerencia de French de que cualquiera que quisiera donar un dólar para cubrir los gastos de notificación y boletín de noticias había recaudado 52,63 dólares en la tercera reunión) y no había elecciones de directivos.

En la cuarta reunión, estaba claro que el Homebrew Computer Club iba a ser un paraíso para los hackers. Más de cien personas recibieron el correo, que anunciaba que la reunión se celebraría esa semana en la Peninsula School, una escuela privada aislada enclavada en una zona boscosa de Menlo Park.

Para entonces, Steve Dompier ya había construido su Altair: había recibido el envío final de piezas a las 10 de la mañana y pasó las siguientes treinta horas armándolo, sólo para descubrir que la memoria de 256 bytes no funcionaba. Seis horas más tarde se dio cuenta de que el fallo se debía a un rasguño en un circuito impreso. Lo arregló cableando la placa y luego trató de averiguar qué hacer con ella.

Parece que la única opción proporcionada por MITS para aquellos que realmente terminaron de construir la máquina era un programa en lenguaje de máquina que se podía ingresar en la máquina solo mediante la fila de pequeños interruptores en el panel frontal. Era un programa que usaba las instrucciones del chip 8080 LDA, MOV, ADD, STA y JMP. Si todo estaba bien, el programa sumaría dos números. Se podría saber traduciendo mentalmente el código de los LED parpadeantes de su forma octal a un número decimal regular. Te sentirías como el primer hombre que pisó la luna, una figura en la historia: tendrías la respuesta a la pregunta que desconcertó a la humanidad durante siglos: ¿Qué sucede cuando sumas seis y dos? ¡Ocho! "Para un ingeniero que aprecia las computadoras, ese fue un evento emocionante", diría más tarde el propietario de Altair y miembro del Homebrew Club Harry Garland, admitiendo que "podría resultar difícil explicarle a un extraño por qué fue emocionante". Para Steve Dompier fue emocionante.

No se detuvo allí. Hizo pequeños programas en lenguaje de máquina para probar todas las funciones de los chips (tenían que ser pequeños programas, ya que la memoria del Altair era minúscula). Hizo esto hasta que sus diez “dispositivos de entrada” (sus dedos) tuvieron callos gruesos. El chip 8080 tenía un conjunto de instrucciones de 72 funciones, así que había mucho por hacer. Dompier, un piloto aficionado, escuchó una radio de baja frecuencia que transmitía el clima mientras trabajaba y, después de probar un programa para ordenar algunos números, sucedió algo muy extraño cuando presionó el interruptor para “ejecutar” el programa: la radio comenzó a hacer ruidos ¡ZIPPPP! ¡ZIIIP! ¡ZIIIIIIIPPPP! Al parecer, estaba reaccionando a la interferencia de radiofrecuencia causada por el cambio de bits de una ubicación a otra dentro del Altair. Acercó la radio y ejecutó el programa nuevamente. Esta vez los ZIP eran más fuertes. Dompier estaba exultante: había descubierto el primer dispositivo de entrada/salida para la computadora Altair 8800.

Ahora la idea era controlar el dispositivo. Dompier trajo su guitarra y descubrió que uno de los ruidos que hacía el ordenador (en la dirección de memoria 075) era equivalente a un fa sostenido en la guitarra. Así que se puso a programar hasta que descubrió las posiciones de memoria de otras notas. Después de unas ocho horas, había trazado la escala musical y escrito un programa para escribir música. Aunque era un programa sencillo, nada parecido al elegante programa musical de Peter Samson en el PDP-1, a Dompier le llevó un tiempo infernal (y doloroso) introducirlo mediante esos interruptores enloquecedores. Pero estaba listo con su interpretación de "Fool on the Hill" de los Beatles (la primera partitura que encontró) para la reunión de Homebrew en la Peninsula School.

La reunión se celebró en una sala del segundo piso de la escuela, un enorme y antiguo edificio de madera que parecía sacado de La familia Addams. El Altair de Dompier era, por supuesto, objeto de mucha adoración, y se moría de ganas de mostrarles la primera aplicación documentada. Pero cuando Dompier intentó encender el Altair, no funcionó. La toma de corriente eléctrica no funcionaba. La toma de corriente más cercana que funcionaba estaba en el primer piso del edificio y, tras localizar un cable de extensión lo suficientemente largo como para llegar hasta el segundo piso, Dompier finalmente logró enchufar su Altair, aunque el cable no era lo suficientemente largo y la máquina tuvo que estar un poco fuera de la puerta. Dompier comenzó el largo proceso de pulsar los interruptores correctos para ingresar la canción en código octal y estaba a punto de terminar cuando dos niños que habían estado jugando en el pasillo tropezaron accidentalmente con el cable y lo sacaron de la pared. Esto borró el contenido de la memoria de la computadora, que Dompier había estado ingresando poco a poco. Comenzó de nuevo y finalmente hizo callar a todos para preparar la primera demostración pública de una aplicación Altair en funcionamiento.

Presionó el interruptor de RUN.

La pequeña radio en la parte superior de la gran y amenazante caja de la computadora comenzó a hacer ruidos roncos y zumbantes. Era en cierto modo música, y para el momento en que concluyeron los primeros compases de la balada de Paul McCartney, la sala de hackers, normalmente llena de chismes sobre el último chip, se sumió en un silencio sobrecogido. El ordenador de Steve Dompier, con la inocencia pura y estremecedora del primer recital de un alumno de primer grado, estaba reproduciendo una canción. En cuanto sonó la última nota, hubo un silencio total y atónito. Acababan de oír pruebas de que el sueño que habían estado compartiendo era real. Un sueño que sólo unas semanas antes les había parecido vago y lejano.

Mucho antes de que tuvieran la oportunidad de recuperarse... el Altair empezó a sonar de nuevo. Nadie (excepto Dompier) estaba preparado para esta repetición, una versión de Daisy, que algunos de ellos sabían que era la primera canción que se tocó en un ordenador, en Bell Labs en 1957; ese acontecimiento trascendental en la historia de la informática se estaba reproduciendo ante sus oídos. Fue un bis tan inesperado que parecía provenir de la conexión genética de la máquina con sus antepasados gigantes (una noción aparentemente implícita en 2001 de Kubrick, cuando la computadora HAL, al ser desmantelada, retrocedió a una versión infantil de esa misma canción).

Cuando el Altair terminó, el silencio no duró mucho. La sala estalló en aplausos y vítores salvajes, los hackers se pusieron de pie de un salto mientras se golpeaban las manos. La gente de Homebrew era una mezcla de profesionales demasiado apasionados como para dejar la informática en sus trabajos, aficionados paralizados por las posibilidades de la tecnología y guerrilleros tecnoculturales dedicados a derrocar una sociedad opresora en la que el gobierno, las empresas y especialmente IBM habían relegado las computadoras a un sacerdocio despreciado. Lee Felsenstein los llamaría "un grupo de fugitivos, al menos fugitivos temporales de la industria, y de alguna manera los jefes no estaban mirando. Y nos juntamos y empezamos a hacer cosas que no importaban porque no era lo que hacían los grandes. Pero sabíamos que ésta era nuestra oportunidad de hacer algo de la manera en que pensábamos que debía hacerse". Esto implicó nada menos que una reescritura importante de la historia de la informática, y de alguna manera este pequeño y sencillo recital musical del Altair de Steve Dompier parecía el primer paso. "Fue un logro importante en la historia de la informática, en mi opinión", dijo Bob Marsh más tarde. Dompier escribió sobre la experiencia, junto con el código en lenguaje de máquina para el programa, en el siguiente número de PCC bajo el título "Música, de algún tipo", y durante meses después los propietarios de Altair lo llamaban en mitad de la noche, a veces tres a la vez en conferencias telefónicas, tocándole fugas de Bach.

Dompier recibió más de cuatrocientas llamadas de ese tipo. Había muchos más hackers de los que nadie imaginaba.

• • • • • • • •

Bob Marsh, el compañero de garaje desempleado de Lee Felsenstein, salió de la primera reunión de Homebrew casi aturdido por la emoción de lo que había hecho en ese pequeño garaje. Sabía que hasta ahora sólo un pequeño número de personas se habían atrevido a concebir el acto de la informática personal. Ahora ahí estaba el melenudo Steve Dompier diciendo que esa compañía al azar, MITS, tenía miles de pedidos. Bob Marsh se dio cuenta en ese mismo momento de que la hermandad de hackers iba a crecer exponencialmente en los próximos años. Pero como un fuego furioso, necesitaba combustible. Los LED parpadeantes del Altair eran emocionantes, pero sabía que, como los hackers son hackers, habría una demanda de todo tipo de dispositivos periféricos, dispositivos que esta compañía MITS obviamente no podría proporcionar.

Pero alguien tendría que hacerlo, porque el Altair era la base de un sistema fantástico para construir nuevos sistemas, nuevos mundos. De la misma manera que el PDP-1 o el PDP-6 habían llegado al MIT como una caja mágica sin un sistema operativo satisfactorio, y de la misma manera que los hackers del MIT le habían proporcionado ensambladores, depuradores y todo tipo de herramientas de hardware y software para que fuera útil en la creación de nuevos sistemas e incluso algunas aplicaciones, dependía de estos hackers de hardware, todavía desorganizados, dejar su propia huella en el Altair 8800.

Bob Marsh comprendió que este era el comienzo de una nueva era y una oportunidad estupenda. Sentado en el frío suelo del garaje de Gordon French, decidió que diseñaría y construiría algunas placas de circuitos que se conectarían a una de las ranuras vacías del bus Altair.

Bob Marsh no era el único que tenía esa idea. De hecho, allí mismo, en Palo Alto (la ciudad próxima a Menlo Park, donde se celebraba la reunión), dos profesores de Stanford llamados Harry Garland y Roger Melen ya estaban trabajando en placas complementarias para el Altair. No habían oído hablar de la reunión, pero asistirían a la segunda reunión de entusiastas del hardware y, a partir de entonces, serían asistentes habituales.

Los dos doctores habían oído hablar por primera vez del Altair cuando Melen, un hombre alto y corpulento cuyo ingenio sólo se veía ligeramente afectado por un tartamudeo recurrente, visitó a Les Solomon a finales de 1974 en la oficina de Nueva York de Popular Electronics. Melen y Garland habían escrito artículos en los que describían proyectos de aficionados para la revista en su tiempo libre y estaban preparando un artículo que contaba cómo construir un dispositivo de control de cámara de televisión.

Melen notó una caja extraña en el escritorio de Solomon y le preguntó qué era. Solomon le informó que la caja, el prototipo de Altair que Ed Roberts había enviado para reemplazar al que se perdió en un flete aéreo, era un microordenador 8080 que se vendía por menos de cuatrocientos dólares. Roger Melen no creía que tal cosa fuera posible, y Les Solomon le dijo que si lo dudaba, debería llamar a Ed Roberts en Albuquerque. Melen lo hizo sin dudarlo y arregló una escala en su camino de regreso al Oeste. Quería comprar dos de esos ordenadores. Además, Ed Roberts había licenciado previamente un proyecto sobre el que Melen y Garland habían escrito en Popular Electronics y nunca había llegado a pagarles regalías. Así que había dos cosas de las que Melen quería hablar con Roberts.

La computadora Altair era la más importante con diferencia -el juguete adecuado en el momento adecuado, pensó Melen- y estaba tan entusiasmado con la perspectiva de tener uno que no pudo dormir esa noche. Cuando finalmente llegó a la modesta sede del MITS, se sintió decepcionado al descubrir que no había ningún Altair listo para llevarse a casa. Pero Ed Roberts era un tipo fascinante, un ingeniero de pura cepa con una visión brillante. Hablaron hasta las cinco de la mañana sobre los aspectos técnicos de esta visión. Sin embargo, esto fue antes de que saliera el artículo de Popular Electronics, y Roberts estaba preocupado por la respuesta que podría recibir. Pensó que no estaría mal que alguien fabricara placas para poner en el Altair y hacerlo útil, y aceptó enviar a Melen y Garland un primer prototipo, para que pudieran hacer algo para conectar una cámara de televisión a la máquina, y luego una placa para emitir una imagen de vídeo también.

Así que Garland y Melen se pusieron manos a la obra y bautizaron su empresa Cromemco, en honor a la residencia de estudiantes de Stanford en la que habían vivido una vez, Crowthers Memorial. Estaban encantados de encontrar espíritus similares en el Homebrew Club, entre ellos Marsh, que había convencido a su amigo Gary Ingram para que los ayudara a fundar una empresa llamada Processor Technology.

Marsh sabía que la mayor necesidad inmediata de un propietario de una Altair era una memoria mayor que los miserables 256 bytes que venían con la máquina, así que pensó que fabricaría una placa que proporcionara 2K de memoria. (Cada "K" equivale a 1.024 bytes). MITS había anunciado sus propias placas de memoria y había entregado algunas a los clientes. Eran bellas placas de memoria, pero no funcionaban. Marsh tomó prestada la Altair de PCC y la examinó detenidamente,

leyendo el manual de principio a fin. Esto era una necesidad porque inicialmente no podía permitirse gastar el dinero para hacer una fotocopia. Pensó que dirigiría la empresa de la misma manera que Roberts aparentemente dirigía MITS: anunciando primero su producto y luego recaudando el dinero necesario para diseñar y fabricarlo.

Así que el Día de los Inocentes, Marsh e Ingram, un ingeniero solitario que no asistía a las reuniones de Homebrew ("No era el tipo de cosas que hacía", explicó Marsh más tarde), inauguraron oficialmente la empresa. Marsh logró reunir el dinero suficiente para fotocopiar cincuenta volantes que explicaban la línea de productos propuestos. El 2 de abril, Marsh se puso de pie en la tercera reunión de Homebrew, repartió los volantes y anunció un descuento del veinte por ciento para cualquiera que hiciera un pedido por adelantado. Después de una semana, no había oído nada. Como Marsh dijo más tarde, "La desesperación se apoderó de nosotros. Pensamos, la hemos cagado, no va a funcionar. Entonces llegó nuestro primer pedido, para una placa ROM [memoria] que costaba sólo cuarenta y cinco dólares. Una orden de compra que pedía "Condiciones de pago a 30 días netos", de una empresa llamada Cromemco. Pensamos, "¿Quién es este Cromemco? ¿Y por qué no pagan en efectivo?" La desesperación se apoderó de nosotros una vez más. ¡NO VA A VOLAR! Al día siguiente llegaron tres pedidos, y en una semana después de eso teníamos dos mil quinientos dólares en efectivo. Recibimos mil, pagamos un anuncio de sexta página en Popular Electronics, y después de eso se desató el infierno. Nos tomó sólo dos meses conseguir cien mil dólares en pedidos".

La ironía era que Marsh y las otras operaciones dirigidas por hackers no se estaban preparando para convertirse en grandes empresas. Buscaban una forma de financiar su afición a jugar con la electrónica, a explorar este nuevo reino de las pequeñas computadoras. Para Marsh y los demás que abandonaron las primeras reuniones de Homebrew con fervor por la construcción de placas, la diversión estaba empezando: diseñar y construir cosas, expresándose mediante los giros y enredos de una placa de circuito integrado de lógica digital para ser conectada al bizantino bus de Ed Roberts.

Como descubrió Marsh, construir una placa para el Altair era el equivalente para un hacker de Homebrew a intentar escribir una gran novela. Sería algo que los críticos severos de Homebrew examinarían cuidadosamente, y no sólo notarían si funcionaba o no, sino que juzgarían la relativa belleza y estabilidad de su arquitectura. La disposición de los circuitos en la placa era una ventana a la personalidad del diseñador, e incluso detalles superficiales como la calidad de los agujeros por los que se montaba la placa delataban los motivos del diseñador, su filosofía y su compromiso con la elegancia. Los diseños digitales, como los de las computadoras, así como los programas de computadora - a decir de Lee Felsenstein - "son las mejores imágenes de mentes que se pueden obtener". "Hay cosas que puedo decir sobre las personas a partir de los diseños de hardware que veo. Puedes mirar algo y decir: 'Dios mío, este tipo diseña como una lombriz de tierra: va de un lugar al final y ni siquiera sabe qué fue lo que hizo en el medio'".

Bob Marsh quería que Processor Technology fuera conocida por sus productos de calidad, y pasó los siguientes meses en un estado de agotamiento, tratando no solo de terminar sus proyectos, sino de hacerlos bien. Era importante para la empresa y también para su orgullo.

El proceso no era terriblemente simple. Después de averiguar qué haría tu placa, pasarías largas noches diseñando el diseño. Mirando el manual que describía el funcionamiento del chip 8080, anotarías los números de las diversas secciones que querías (designando esta sección para una

entrada, esa para la memoria) y la cuadrícula laberíntica dentro de esa pieza de plástico negro comenzaría a remodelarse dentro de tu cabeza. La eficacia de la elección de las secciones a las que acceder dependería de lo bien y con qué precisión mantuvieras esa visión. Dibujarías a lápiz esas conexiones, con lo que iría en un lado de la placa escrito en azul, lo que iría en el otro lado en rojo. Luego conseguirías hojas de Mylar, las colocarías sobre una cuadrícula en una mesa de luz y comenzarías a trazar el contorno de las conexiones, usando cinta de papel crepé. Podrías descubrir que tu esquema tenía algunos problemas (demasiado tráfico en una parte, las interconexiones demasiado apretadas) y tendrías que realinear algunas cosas. Un error podría arruinarlo todo. Así que te asegurarías de hacer una superposición del esquema: colocándola sobre tu diseño pegado con cinta, podrías ver si cometiste algún error grave, como conectar tres cosas juntas. Si el esquema en sí tenía un error, olvídate.

Lo diseñarías de modo que la placa tuviera varias capas; un conjunto diferente de conexiones en la parte superior y la inferior. Mientras trabajabas, dabas vueltas al diseño y, a veces, la cinta se despegaba, o te sobraban trocitos de cinta, o se te quedaba un pelo atascado en alguna parte: cualquiera de estos fenómenos innecesarios se reproducía fielmente en las reproducciones en sepia que te hacían en una casa de línea azul (si no tenías dinero para eso, hacías una cuidadosa fotocopia) y el resultado era un cortocircuito desastroso. Después, marcabas el diseño para la empresa de placas, indicando dónde perforar y qué necesitaba chapado en oro, etc.

Finalmente, ibas a una empresa de placas local con los planos en la mano. Se los entregabas. Como todavía había recesión, se alegraban por el negocio, incluso si venía de un hacker de hardware desaliñado, de poca monta y con los ojos vidriosos. Colocaban el dispositivo en un digitalizador, perforaban los agujeros y producían sobre un material epoxi verdoso un lío de interconexiones plateadas. Ese era el método de lujo. Al principio, Bob Marsh no podía permitírselo, así que grabó a mano la placa sobre la estufa de la cocina, utilizando material laminado de circuito impreso, creando líneas apenas perceptibles en las que se fundiría el material. Ese método fue un tortuoso cortejo a la diosa perra Desastre, pero Marsh era un trabajador compulsivamente cuidadoso. Más tarde explicó: "Realmente me meto en ello. Me convierto en uno con mi diseño esquemático".

Para esta primera placa de memoria, Marsh se encontró bajo una presión particular. Cada dos semanas en la reunión de Homebrew, todos los días en el teléfono, la gente frenética buscaba placas de memoria estáticas como buzos que jadean por aire. Marsh recordó más tarde sus gritos: "¿Dónde está mi placa? La necesito. TENGO QUE TENERLA".

Finalmente, Marsh terminó. No había tiempo para un prototipo. Tenía su placa, que era el rectángulo de epoxi verde con una pequeña protuberancia de conectores dorados grabados debajo, dimensionada para encajar en una ranura en el bus Altair. Tenía los chips y los cables que los constructores de kits soldarían en él (al principio, Processor Tech solo vendía placas sin ensamblar). Marsh lo tenía todo listo, y no tenía un Altair para probarlo. Así que, a pesar de que eran las tres de la mañana, llamó a ese tipo Dompier que conocía de Homebrew y le dijo que trajera la máquina. El Altair de Dompier era al menos tan valioso para él como lo sería un bebé humano si no estuviera en modo soltero, así que lo envolvió cuidadosamente en una pequeña manta roja para llevárselo. Dompier había seguido las reglas al ensamblar la máquina, incluso se puso una pulsera de cobre alrededor de la muñeca cuando soldaba (para mantener baja la electricidad estática) y tuvo cuidado de no tocar el frágil corazón 8080 de la máquina. Así que se quedó atónito, después de dejar con cariño la máquina en el taller de Marsh, cuando los veteranos del hardware Marsh e Ingram comenzaron a manipular los chips como un par de mecánicos de garaje instalando un silenciador.

Agarraban fichas con sus dedos sucios y las tiraban por todos lados, las sacaban y las volvían a meter. Dompier observaba horrorizado. Finalmente, tenían la placa lista, e Ingram encendió el interruptor y el preciado ordenador de Steve Dompier se quedó inconsciente. Habían puesto la placa al revés.

Llevó un día arreglar el Altair de Dompier, pero Steve Dompier no albergaba ningún enojo: de hecho, prestó su máquina a Processor Technology para futuras pruebas. Era un indicio de comportamiento Homebrew. Estos eran una raza de hackers diferente a los inaccesibles magos del MIT, pero todavía se aferraban a la Ética Hacker que sublimaba la posesión y el egoísmo en favor del bien común, lo que significaba cualquier cosa que pudiera ayudar a la gente a hackear de manera más eficiente. Steve Dompier estaba nervioso por su Altair, pero quería poco en el mundo más que una placa de memoria para poder ejecutar algunos programas reales en la máquina. Y luego quería dispositivos de E/S, dispositivos de visualización... para poder escribir utilidades para hacer la máquina más poderosa. Herramientas para hacer herramientas, para adentrarse en el mundo que se centraba en el misterioso microprocesador 8080 dentro de su máquina. Bob Marsh y los demás en Homebrew, ya sea que ofrecieran productos para la venta o fueran simplemente hackers curiosos como él, estaban todos juntos en esto, y juntos formaron una comunidad que puede no haber estado tan centrada geográficamente como la comunidad PDP-6 del MIT (se extendía desde Sacramento hasta San José), pero que de todos modos estaba fuertemente unida.

Cuando Bob Marsh se presentó en una reunión de Homebrew a principios de junio con el primer envío de placas, las personas que las habían pedido estaban tan agradecidas que podría pensarse que las había estado regalando. Entregó los pequeños paquetes de plástico envueltos en blíster con la placa y los circuitos integrados, junto con el manual de instrucciones que Lee Felsenstein había escrito. "A menos que seas un constructor de kits experimentado", advirtió Lee, "no construyas este kit".

Había muy poca experiencia en el mundo en la construcción de ese tipo de cosas, pero gran parte de la experiencia que existía en el mundo se centraba en esa sala de reuniones, que ahora era el auditorio del Acelerador Lineal de Stanford (SLAC). Habían pasado cuatro meses desde la primera reunión informal del club y el número de miembros se había multiplicado por diez.

• • • • • • • •

El pequeño club formado por Fred Moore y Gordon French había crecido hasta convertirse en algo que ninguno de los dos podría haber imaginado. Era la vanguardia de una raza de hackers de hardware que se estaban “esforzando” por entrar en una nueva industria que, estaban seguros, sería diferente de cualquier industria anterior. La industria de las microcomputadoras estaría regida por la ética hacker. (El término "bootstrap" era indicativo de la nueva jerga hablada por estos hackers: el término describe literalmente el proceso por el cual un programa de computadora se alimenta a sí mismo en una máquina cuando la máquina se enciende por primera vez, o "arranca". Parte del programa alimentará el código a la computadora; este código programará a la máquina para que se diga a sí misma que debe alimentar el resto del código. Es como levantarse por los propios medios. Es un símbolo de lo que la gente del Homebrew estaba haciendo: crear un nicho en el mundo de los pequeños sistemas informáticos, y luego cavar más profundo para convertir el nicho en una caverna, un asentamiento permanente.)

Pero los fundadores del club pronto se vieron superados por la brillantez técnica que los rodeaba. En el caso de French, sufría de lo que parecía ser una actitud burocrática latente. En algunos

aspectos, su manía por mantener el progreso del club de una manera ordenada y controlada fue útil. Actuó como secretario y bibliotecario, manteniendo una lista de los números de teléfono de todos y el equipo que poseía cada uno. Como recordó más tarde, "Mi teléfono sonaba sin parar. Fue increíble. Todo el mundo necesitaba información y se necesitaban unos a otros para ponerse en marcha porque había una absoluta escasez de equipos. Por ejemplo: 'Si tienes un terminal, ¿me lo podrías prestar durante un par de días mientras le pongo mi programa para que lea mi lector de cinta de papel perforado?' Ese tipo de cosas".

Pero en otros aspectos, especialmente en la forma en que moderaba las reuniones, el estilo de French no era coherente con el espíritu hacker que se estaba gestando en Homebrew. "Gordon era un tipo didáctico", recordaría más tarde Lee Felsenstein. "Intentaba llevar la discusión hacia donde él quería que fuera. Quería que fuera un evento educativo, dando conferencias, enseñando a la gente sobre ciertas cosas, especialmente sobre cosas en las que él era experto. Se molestaba mucho si la discusión se desviaba de lo que la gente literalmente enseñaba a otras personas en un sentido escolar. Él se involucraba en lo que la gente decía y se involucraba en el contenido, inyectando sus opiniones y diciéndoles: "Hay un punto importante que no se debe pasar por alto, y sé más sobre este tipo de cosas". Después de la primera parte de la reunión, en la que las personas se presentaban y decían en qué estaban trabajando, Gordon se paraba frente a la sala y daba lo que equivalía a un tutorial, explicando la forma en que la máquina usa el código que le ingresas e informando a los miembros inquietos cómo aprender buenos hábitos de codificación te ahorrará dolores de cabeza en el futuro... y tarde o temprano la gente se impacientaba tanto que se escapaba de las reuniones y comenzaba a intercambiar ideas e información en la sala. Era una situación delicada, el tipo de dilema humano complejo que a los hackers no les gusta afrontar en general. Pero surgió la sensación de que un nuevo moderador debería tomar el relevo.

La elección lógica podría haber sido Fred Moore, que se sentó en la parte delantera de la sala durante los primeros meses de Homebrew con su grabadora y su cuaderno, grabando las reuniones para poder resumir los puntos destacados en el boletín que publicaba todos los meses. Estaba dedicando mucho de su tiempo al grupo, porque vio que los hackers y sus Altairs estaban al borde de lo que podría ser una fuerza social significativa. "Al compartir nuestra experiencia e intercambiar consejos, avanzamos en el estado del arte y hacemos posible la informática de bajo coste para más gente", escribió en el boletín, añadiendo su comentario social: "La evidencia es abrumadora de que la gente quiere ordenadores, probablemente para su propio entretenimiento y uso educativo. ¿Por qué las grandes empresas perdieron este mercado? Estaban ocupadas vendiéndose máquinas a precios excesivos entre sí (y al gobierno y al ejército). No quieren vender directamente al público. Estoy totalmente a favor del revuelo que MITS está teniendo con el Altair porque hará tres cosas: (1) obligar a otras empresas a despertar ante la demanda de ordenadores de bajo coste para el hogar... (2) hacer que se formen clubes informáticos locales y grupos de aficionados para llenar el vacío de conocimiento técnico, (3) ayudar a desmitificar las computadoras... ”

Moore identificó explícitamente el propósito del club como un intercambio de información. Al igual que el flujo sin trabas de bits en un ordenador de diseño elegante, la información debería circular libremente entre los participantes en Homebrew. “Más que cualquier otro individuo, Fred Moore sabía de qué se trataba compartir”, recordó más tarde Gordon French. “Esa era una de las expresiones que siempre utilizaba: compartir, compartir, compartir”.

Pero la mayoría del club prefería un camino que divergía del de Fred Moore. Fred siempre insistía en las aplicaciones. De vez en cuando, en las primeras reuniones, instaba a los miembros de este

grupo básicamente anarquista a reunirse y hacer algo, aunque normalmente era vago sobre lo que podría ser ese algo.

Tal vez usar computadoras para ayudar a personas discapacitadas, tal vez compilar listas de correo para la resistencia al reclutamiento. Moore podría haber estado en lo cierto al percibir que el impulso del club era en cierto modo político, pero su punto de vista parecía estar en desacuerdo con la realidad de que los hackers no suelen proponerse crear cambios sociales: los hackers actúan como hackers. Moore estaba menos fascinado con el funcionamiento de los sistemas informáticos que con la idea de generar un sistema social benévolo y de compartir; parecía considerar Homebrew no como un bastión técnico de gente ávida de la capacidad de construcción de pirámides de los ordenadores domésticos, sino como un grupo dedicado al cambio social, como la resistencia al reclutamiento o los grupos antinucleares en los que había estado involucrado. Sugería la venta de pasteles para recaudar fondos para el grupo, o publicaba poemas simpáticos en el boletín como "No te quejes ni te quejes / Depende de cada uno de nosotros / Hacer que el Club haga / Lo que queremos que haga". Mientras tanto, la mayoría de los miembros del club estarían consultando la parte de atrás del boletín para estudiar los esquemas de la contribución titulada “Generación de funciones lógicas arbitrarias mediante multiplexores digitales”. Esa era la manera de cambiar el mundo, y mucho más divertida que una venta de pasteles.

Lee Felsenstein reflexionó más tarde que no creía que Moore “tuviese clara su política. En el nivel superficial se quedaba en el punto de la protesta o el gesto de protesta. Pero estábamos mucho más interesados en lo que podríamos llamar la propaganda de los hechos”.

Así que cuando apareció fortuitamente una oportunidad para hacer que las reuniones fuesen más compatibles con el espíritu hacker que fluía libremente (Gordon French, que hacía trabajos de consultoría para la Administración de la Seguridad Social, fue llamado temporalmente a Baltimore), algunos miembros del club no pidieron a Moore que moderase, sino a Lee Felsenstein. Resultó ser una elección ideal, ya que era tan hacker de hardware como cualquiera, pero también un informático político. Consideró que la llamada a moderar estas reuniones era un ascenso significativo. Ahora podía ser el hombre clave de la revolución en el frente del hardware, permitiendo que las reuniones progresaran con la combinación justa de anarquismo y dirección, continuando con sus propios esquemas de diseño de hardware de guerrilla que llevarían al triunfo de la Terminal Tom Swift, y participando en la resurrección del concepto latente de Memoria Comunitaria, un proceso que estaba comenzando ese verano con la publicación de una revista mimeografiada llamada Journal of Community Communications, que difundiría el concepto de dispositivos de microcomputadoras "creados y utilizados por personas en sus vidas diarias como miembros de comunidades".

Sin embargo, cuando se paró por primera vez frente a la sala en una reunión de Homebrew en junio de 1975, estaba aterrorizado. Según recuerda, alguien preguntó quién sería el nuevo moderador, y Marty Spergel, el "hombre de la chatarra" que era dueño de la casa de suministros electrónicos M&R, sugirió a Lee, y "el clamor se disparó". Fue como si hubiera sido coronado. A pesar de lo nervioso que estaba, era una oportunidad que no podía dejar pasar. Como era habitual en Lee, los riesgos de fracasar eran menos intimidantes que los riesgos de no intentarlo en absoluto.

Sabía un poco sobre cómo dirigir un foro. Durante sus días de estudiante radical en 1968, había estado escuchando un programa de radio de Berkeley con llamadas entrantes que estaba tan mal diseñado, con participantes inaudibles y distorsión y cosas así, que corrió al estudio agitando su

radio portátil y diciendo: "¡Escuchen esto, idiotas!". Terminó ayudando a dirigir el programa, y parte de su papel era preparar a los invitados antes de que salieran al aire. Pensó que su papel en Homebrew podría aprovechar eso; instó a las personas que no estaban acostumbradas a dirigirse a una audiencia más grande que una mesa llena de piezas electrónicas a hablar con otros humanos sobre sus intereses. Como intuyó Fred Moore, este iba a ser el corazón de la reunión, el intercambio de información. Lee, creando una arquitectura para la reunión como si estuviera abordando un problema de diseño electrónico, hizo un diagrama de flujo de la sesión. Habría un momento para recorrer la sala y dejar que la gente dijera lo que estaba haciendo o lo que quería saber (esa sería la sección de “mapeo”, similar a dibujar un esquema). Luego habría una sección de “acceso aleatorio”, donde uno se dirigiría a las personas que sugirieran cosas que le interesaran, o que pudieran responder a sus preguntas, o que parecieran tener la información que deseaba, o simplemente pareciera interesante hablar con ellas. Después de eso, tal vez habría una breve charla, o alguien demostraría un sistema o mostraría un nuevo producto, y luego habría más mapeo y más acceso aleatorio. Cuando Lee vio que la gente se mostraba reacia a regresar de la primera sección de acceso aleatorio (a veces uno podía perderse en algún punto técnico o en algún asunto religioso como una técnica para enrollar un tablero o algo así), cambió la estructura para incluir solo una sección de acceso aleatorio, al final de la reunión. Así depurada, la estructura funcionó bien.

Lee descubrió que estar frente a un grupo de personas que lo aceptaban y apreciaban su rol como “puntero de pila” (la función de la computadora que determina el orden en que se realizan las tareas computacionales) lo ayudaba en su esfuerzo consciente por salir de su caparazón de topo. Poco después de comenzar su mandato como moderador, se sintió lo suficientemente seguro como para dar una charla al grupo en su terminal Tom Swift; garabateando en la pizarra al frente del pequeño auditorio de SLAC, habló sobre pantallas de video, confiabilidad del hardware, Ivan Illich y la idea de incorporar al usuario en el diseño. Fue una buena mezcla de comentario social y esoterismo técnico, y los Homebrewers lo apreciaron. Lee descubrió que tenía talento para la ocurrencia rápida y finalmente hackeó una pequeña rutina que pronunciaba al comienzo de cada reunión. Llegó a enorgullecerse ferozmente de su trabajo como maestro de ceremonias del club: en su pensamiento, ahora era el maestro de ceremonias de un movimiento hacker, un grupo que era central en la configuración de un estilo de vida basado en microprocesador.

Poco después de que Lee asumiera el cargo, un Fred Moore preocupado renunció a sus funciones de tesorero, secretario y editor del boletín. Tenía algunos problemas personales; la mujer con la que había estado saliendo lo había dejado. Fue un momento difícil para él al irse: sentía que el club había sido su legado, en cierto sentido, pero probablemente para entonces estaba claro que sus esperanzas de que se dedicara a un trabajo de servicio público eran inútiles. En cambio, estaba la Propaganda de la Escritura y, lo que era más inquietante, algunas personas que acudían a las reuniones, según recordó Fred más tarde, “con carteles de billetes de dólar en los ojos, diciendo: ‘Vaya, aquí hay una nueva industria, voy a construir esta empresa y fabricar estas placas, y ganar un millón...’”. Había otros temas sociales relacionados con la informática que Moore quería abordar, pero se había dado cuenta, según explicó más tarde, de que “la gente del club estaba muy por delante [de él] en cuanto a conocimientos de electrónica o informática, [y debido a esto] la gente estaba enamorada de esos mismos dispositivos, dispositivos que eran muy seductores”. Fred estaba descontento con la forma tan ciega en que la gente aceptaba la tecnología. Alguien le había hablado de la mano de obra barata que había en Malasia y otros países asiáticos para ensamblar físicamente esos chips mágicos. Había oído que a las mujeres asiáticas se les pagaban salarios miserables, trabajaban en fábricas inseguras y no podían regresar a sus aldeas, ya que nunca habían tenido la

oportunidad de aprender los métodos tradicionales de cocinar o criar una familia. Pensó que debía contárselo al club, forzar el asunto, pero para entonces se dio cuenta de que no era el tipo de problema que el Homebrew Club se suponía que debía abordar.

Aun así, amaba al club, y cuando sus problemas personales lo obligaron a retirarse y regresar al Este, diría más tarde que fue "uno de los días más tristes de mi vida". Una figura pequeña y melancólica, se paró frente a la pizarra en una reunión de mediados de agosto y escribió sus deberes, preguntando quién haría el boletín, quién se encargaría de la tesorería, las notas... Y alguien se acercó y comenzó a escribir "Fred Moore" junto a cada ítem. Le partía el corazón, pero sentía que por él se había acabado, y aunque no podía compartir todas sus razones, tenía que hacerles saber a sus hermanos que ya no podía estar allí.

“Me veía como una persona que había ayudado a esas personas a reunirse y compartir sus habilidades y energía”, dijo Moore más tarde. Y esas metas se habían alcanzado. De hecho, cada reunión parecía rebosar de espíritu y entusiasmo mientras la gente intercambiaba chismes y chips, abriéndose paso en este nuevo mundo. En el período de mapeo, la gente se ponía de pie y decía que tenían un problema para configurar esta o aquella parte del Altair, y Lee preguntaba: “¿Quién puede ayudar a este tipo?” y tres o cuatro manos se levantaban. Bien. ¿Siguiente? Y alguien decía que necesitaba un chip 1702. Alguien más podría tener un chip 6500 adicional, y se haría un intercambio.

Luego había gente que se ponía de pie para anunciar los últimos rumores en Silicon Valley. Jim Warren, un corpulento ex estudiante de posgrado en informática de Stanford, era un chismoso particularmente bien conectado que aparecía en el período de acceso aleatorio y hablaba durante diez minutos sobre esta empresa y la siguiente, a menudo incluyendo algunas de sus opiniones personales sobre el futuro de las comunicaciones informáticas mediante transmisiones digitales.

Otro notorio proveedor de esta extraña forma de chisme era un ingeniero novato llamado Dan Sokol, que trabajaba como probador de sistemas en una de las grandes empresas de Silicon Valley. Sus chismes eran a menudo sorprendentemente proféticos (para mantenerlos adivinando, Sokol admitió más tarde, inventaba aproximadamente la mitad de sus rumores). Sokol, un discípulo digital de pelo largo y barba que se lanzó al Homebrew con la energía de los recién convertidos, se adhirió rápidamente a la Ética Hacker. Consideraba que ningún rumor era demasiado clasificado como para compartirlo, y cuanto más importante era el secreto, mayor era su deleite cuando lo revelaba. "¿Hay alguien aquí de Intel?" A veces metía la mano en el bolsillo y sacaba el prototipo de un chip que Intel había estado ocultando exitosamente de cada compañía del Valle (y tal vez de un cuadro del espionaje ruso).

A menudo Sokol, un inveterado regateador, realmente sacaría un chip del bolsillo. Por ejemplo, un día en el trabajo – recordó - unos hombres de una nueva empresa llamada Atari entraron a probar unos chips. Eran extremadamente reservados y no dijeron qué chips eran. Sokol los examinó: algunos estaban marcados como Syntech, otros como AMI. Sokol conocía a gente de ambas empresas y le dijeron que los chips eran piezas personalizadas, diseñadas por la gente de Atari. Así que se llevó uno a casa, lo puso en una placa y lo probó. El chip resultó contener un programa para jugar al nuevo videojuego Pong (la nueva firma Atari estaba empezando a montar un sistema doméstico para jugar a ese juego, en el que dos personas controlan unas “paletas” de luz en una pantalla de televisión e intentan mantener en juego una “pelota” con forma de blip). Sokol trazó el diseño en una placa de circuito, lo llevó a Homebrew y lo mostró. Se llevó unos cuantos chips

adicionales y los intercambió con otros, hasta que acabó con un teclado y unos cuantos chips de RAM. “Estamos hablando de un robo descarado”, explicó más tarde; pero en términos de Homebrew, Sokol estaba liberando un hack genial de los opresores propietarios. Pong era genial y debería pertenecer al mundo. Y en Homebrew, intercambios como ese eran gratuitos y fáciles.

Años antes, Buckminster Fuller había desarrollado el concepto de sinergia (el poder colectivo, más que la suma de las partes, que surge de las personas y/o fenómenos que trabajan juntos en un sistema) y Homebrew era un ejemplo clásico de ese concepto en funcionamiento. La idea de una persona podía impulsar a otra a embarcarse en un gran proyecto y, tal vez, a fundar una empresa para fabricar un producto basado en esa idea. O, si a alguien se le ocurría un truco ingenioso para producir un generador de números aleatorios en el Altair, repartía el código para que todo el mundo pudiera hacerlo y, para la siguiente reunión, alguien más habría ideado un juego que utilizaba la rutina.

La sinergia continuaba incluso después de la reunión, ya que algunos de los aficionados al cómputo casero continuaban sus conversaciones hasta la medianoche en The Oasis, un ruidoso bar cerca del campus (el lugar había sido sugerido por Roger Melen; Jim Warren, un virulento antitabaco, una vez intentó atraer a la gente a la sección para no fumadores de The Village Host, pero eso nunca se hizo popular). Amontonados en cabinas de madera con mesas profundamente grabadas con las iniciales de generaciones de estudiantes de Stanford, Garland, Melen, Marsh, Felsenstein, Dompier, French y todos los demás que tenían ganas de asistir se envalentonaban con la energía de la reunión y las jarras de cerveza. Imaginaron desarrollos tan fantásticos que nadie jamás creería que pudieran ser más que fantasías, fantasías lejanas como el día en que los ordenadores domésticos con pantallas de televisión generarían programas pornográficos —SMUT-ROM, los llamaban— que no serían ilegales porque solo serían pornográficos si los escaneabas como lo hacía el ordenador. ¿Cómo podría ser pornográfico el código informático en bruto? Fue sólo una de las docenas de reflexiones perversamente improbables que no sólo se harían realidad, sino que se superarían en unos pocos años.

Sinergia: Marty Spergel, el Junk Man, sabía exactamente cómo funcionaba eso. Un regateador bronceado, de mediana edad, con una sonrisa encantadoramente amplia, pensaba que Homebrew era como "tener tu propia tropa de Boy Scouts, todos ayudando a todos los demás. Recuerdo que tuve problemas con una máquina de teletipo en mi oficina y un tipo [en Homebrew] dijo que la revisaría. No sólo la revisó, sino que vino con un pequeño kit y puso cuatro o cinco partes diferentes, la engrasó, la lubricó, ajustó todos los engranajes. Le dije: "¿Cuánto te debo?". Dijo: "Nada". Para el Junk Man, esa era la esencia de Homebrew.

Spergel siempre llevaba un registro de las piezas que necesitaba la gente; a veces llevaba una caja de ellas a una reunión. Después de la charla sobre la terminal de Tom Swift, le preguntó a Lee si le interesaba construir una para la empresa de Spergel, M&R Electronics. Bueno, la terminal Swift no estaba lista, le dijo Lee, pero ¿qué tal este diseño para un módem (un dispositivo que permite que las computadoras se comuniquen por las líneas telefónicas) que Lee había hecho un par de años atrás? “Probablemente incluso sabía lo que era un módem, aunque no estaba claro por la forma en que reaccionó”, dijo Lee más tarde. Los módems se vendían entonces por cuatrocientos a seiscientos dólares, pero Marty pudo construir el módem "Pennywhistle" de diseño limpio de Lee para venderlo por $109. Enviaron una copia de los esquemas a Lee Solomon en Popular Electronics y él puso una foto del módem de Lee en la portada.

Sinergia. El creciente número de miembros de Homebrew que estaban diseñando o regalando nuevos productos, desde joysticks de juegos hasta placas de E/S para Altair, utilizaron el club como una fuente de ideas y pedidos tempranos, y para realizar pruebas beta de los prototipos. Cada vez que se terminaba un producto, se lo llevaba al club y se obtenían las críticas más expertas disponibles. Luego se distribuían las especificaciones técnicas y los esquemas; si se trataba de software, se distribuía el código fuente. Todo el mundo podía aprender de él y mejorarlo si quería y era lo suficientemente bueno.

Se creaba un ambiente animado que funcionaba tan bien porque, en consonancia con la ética hacker, no se mantenían límites artificiales. De hecho, todos los principios de esa ética, tal como los habían formado los hackers del MIT, se ponían en práctica en cierta medida dentro de Homebrew. La exploración y las actividades prácticas se reconocían como valores cardinales; la información recogida en estas exploraciones y aventuras de diseño se distribuía libremente incluso a competidores nominales (la idea de la competencia llegó lentamente a estas nuevas empresas, ya que la lucha era crear una versión hacker de una industria, una tarea que requería que todos trabajaran juntos); se desdeñaban las normas autoritarias y la gente creía que las computadoras personales eran los embajadores definitivos de la descentralización; Las filas de miembros estaban abiertas a cualquiera que se acercara, con respeto ganado por la experiencia o las buenas ideas, y no era inusual ver a un joven de diecisiete años conversando como un igual con un ingeniero veterano, próspero y de mediana edad; había un agudo nivel de apreciación de la elegancia técnica y el arte digital; y, sobre todo, estos hackers de hardware veían de una manera vibrantemente diferente y populista cómo las computadoras podían cambiar vidas. Eran máquinas baratas que sabían que estaban a solo unos años de distancia de volverse realmente útiles.

Esto, por supuesto, no les impidió sumergirse por completo en hackear estas máquinas por el mero hecho de hackearlas, por el control, la búsqueda y el sueño. Sus vidas estaban dirigidas a ese momento en el que la placa que diseñaron, o el autobús que conectaron, o el programa que teclearon, haría su primera pasada... Una persona se refirió más tarde a ese momento como algo similar a dar marcha atrás con una locomotora sobre una sección de vía que acabas de arreglar y hacerla pasar por esa vía a noventa millas por hora. Si tu vía no era fuerte, el tren descarrilaba desastrosamente... humo... fuego... metal retorcido... Pero si hackeabas bien, pasaría a toda velocidad con una excitante ráfaga. Te sobresaltarías al darte cuenta de que miles de cálculos por segundo pasarían por ese equipo con tu sello personal. Tú, el maestro de la información y legislador de un nuevo mundo.

Algunos planificadores visitaban Homebrew y se desanimaban por la ferocidad técnica de las discusiones, la llama intensa que ardía más cuando la gente se dedicaba a la búsqueda hacker de la construcción. Ted Nelson, autor de Computer Lib, fue a una reunión y se sintió confundido por todo lo que había, y más tarde llamó a la gente de Homebrew, desaliñados y en gran parte despeinados, “monjes del chip, gente obsesionada con los chips. Era como ir a una reunión de gente a la que le encantan los martillos”. Bob Albrecht rara vez asistía, y más tarde explicó que “sólo podía entender una de cada cuatro palabras que decían esos tipos... eran hackers”. Jude Milhon, la mujer con la que Lee siguió siendo amigo después de su encuentro a través de Barb y su participación en Community Memory, se presentó una vez y se sintió repelida por la concentración en la pura tecnología, la exploración y el control por el mero hecho de controlar. Observó la falta de mujeres hackers de hardware y se enfureció por la obsesión de los hackers masculinos por el juego y el poder tecnológico. Resumió sus sentimientos con el epíteto “los chicos y sus juguetes”, y, como Fred

Moore, le preocupaba que el romance con la tecnología pudiera llevar ciegamente al abuso de esa tecnología.

Ninguna de estas preocupaciones frenó el impulso de Homebrew, que estaba creciendo hasta varios cientos de miembros, llenando el auditorio de SLAC, convirtiéndose en el punto culminante quincenal en las vidas de más de cien Brewers incondicionales. Lo que habían comenzado era casi una cruzada ahora, algo que Ted Nelson, cuyo libro estaba lleno de diatribas contra IBM, debería haber apreciado. Mientras que IBM y los grandes tipos nunca pensaron en estos hackers aleatorios en clubes de computadoras con sus ideas de poseer computadoras, la gente de Homebrew y otros como ellos estaban hackeando no solo los chips 8080, sino los cimientos ahora desmoronados de la Torre Batchprocessed de Bit Babble. “Nos reforzamos mutuamente”, explicó Lee Felsenstein más tarde. Nos brindamos una estructura de apoyo mutuo. Comprábamos los productos de los demás. Nos cubríamos el trasero mutuamente, en efecto. Allí estábamos, la estructura industrial no nos prestaba atención. Sin embargo, teníamos gente que sabía tanto como cualquier otra persona sobre este aspecto de la tecnología, porque era muy nuevo. Podíamos hacer lo que fuera, y lo hicimos”.

• • • • • • • •

Cuando Les Solomon, el gurú neoyorquino de este movimiento, llegó de visita a la Costa Oeste, la época dorada del Homebrew Computer Club estaba brillando en su máximo esplendor. Solomon primero se puso en contacto con Roger Melen y Harry Garland, que acababan de terminar el prototipo del producto de Cromemco que aparecería en la portada de Popular Electronics en noviembre de 1975: una placa adicional para el Altair que permitiría conectar la máquina a un televisor en color, lo que produciría resultados gráficos deslumbrantes. De hecho, Melen y Garland llamaban a la placa “Dazzler”. Les fue al apartamento de Roger para verla, pero antes de poner la placa en el Altair de Roger, los tres se pusieron a beber, y estaban bastante bien lubricados cuando la placa estuvo dentro y el televisor en color encendido.

En aquel momento, había dos programas de Altair que podían sacar partido del Dazzler. Uno se llamaba Kaleidoscope y brillaba y cambiaba de forma. Fue un gran momento para Solomon ver cómo el ordenador que había ayudado a traer al mundo hacía que un televisor en color ejecutara patrones hermosos.

Luego probaron con otro programa: LIFE. The gamethatismorethanagame, creado por el matemático John Conway. El juego que el mago del MIT Bill Gosper había hackeado tan intensamente, hasta el punto de verlo como generador de vida. La versión de Altair funcionaba mucho más lentamente que el programa PDP6, por supuesto, y sin ninguna de esas utilidades elegantemente hackeadas, pero seguía las mismas reglas. Y lo hacía mientras estaba sentado en la mesa de la cocina. Garland introdujo algunos patrones y Les Solomon, que no conocía del todo las reglas del juego y, desde luego, no era consciente de las profundas implicaciones filosóficas y matemáticas, observó cómo las estrellitas azules, rojas o verdes (así era como el Dazzler hacía que se vieran las células) se comían a las otras estrellitas o creaban más estrellas. Qué pérdida de tiempo, pensó. ¿A quién le importa?

Entonces empezó a jugar distraídamente con la máquina, dibujando un patrón para ejecutar. En su estado de ebriedad, colocó algo parecido a la estrella de David. Más tarde recordaría: “Ejecuté el programa y observé cómo se consumía a sí mismo. Tardó unos diez minutos y finalmente murió. Pensé: “Vaya, qué interesante, ¿significa eso que la religión judía está a punto de desaparecer después de doscientas cuarenta y siete generaciones?”. Así que dibujé un crucifijo. Eso duró ciento

veintiuna generaciones. ¿Significa esto de alguna manera que el judaísmo sobrevivirá al cristianismo?". Pronto estaba colocando medialunas y estrellas y símbolos de diferentes significados, y los tres —cuatro de ellos, incluido el Altair— exploraban los misterios de las religiones y nacionalidades del mundo. "¿Quién diablos necesita filosofía a las tres de la mañana, bebiendo?", dijo Solomon más tarde. "Era una computadora. . Estaba allí."

Pero Les Solomon tenía más magia que transmitir. Una de las historias que contaba, historias tan escandalosas que sólo un tacaño de la imaginación se quejaría de su improbabilidad, era la de la época en que estaba explorando en pos de uno de sus "pasatiempos", la arqueología precolombina. Esto requería mucho tiempo en las selvas, "corriendo con los indios, cavando, revolviendo en la tierra… ya sabes, encontrando cosas". Fue de esos indios, insistió Les Solomon, de quienes aprendió el principio vital del vril, un poder que permite mover objetos enormes con muy poca fuerza. Solomon creía que fue el poder del vril lo que permitió a los egipcios construir las pirámides. (Tal vez el vril era el poder del que hablaba Ed Roberts cuando se dio cuenta de que su Altair daría a la gente el poder de diez mil egipcios constructores de pirámides). Según su historia, Solomon conoció a un venerable brujo indio y le preguntó si podía aprender este poder. ¿Podría el brujo enseñarle? Y el brujo obedeció. Ahora, después de la noche de borrachera con el programa LIFE, Solomon asistió a una reunión de Homebrew en SLAC donde se le concedió el respeto de un invitado de honor: la partera de Altair de Ed Roberts. Y después de la reunión, Solomon les contó a los hackers de hardware sobre el Vril. Hubo cierto escepticismo.

Afuera de SLAC había enormes mesas de picnic naranjas con bases de cemento. Solomon hizo que la gente de Homebrew tocara con sus manos una de las mesas, y él también la tocó. Simplemente tenían que pensar que se levantaría.

Lee Felsenstein describió más tarde la escena: "Él dijo: 'Oigan, déjenme mostrarles...' Estábamos pendientes de cada palabra suya, haríamos cualquier cosa. Entonces unas seis personas rodearon la mesa, pusieron sus manos. Él puso su mano encima, entrecerró los ojos y dijo: 'Vamos'. Y la mesa se levantó aproximadamente un pie. Se levantó como un movimiento armónico, [tan elegantemente como] una onda sinusoidal. No se sentía pesada. "Simplemente sucedió".

Después, incluso los participantes, salvo Solomon, ese vieron inseguros de que realmente hubiera sucedido. Pero Lee Felsenstein, al ver que se cerraba otro capítulo de esa trascendental novela de ciencia ficción que era su vida, comprendió el impacto mítico de este evento. Ellos, los soldados del Homebrew Computer Club, habían tomado sus talentos y aplicado la Ética Hacker para trabajar por el bien común. Fue el acto de trabajar juntos al unísono, con las manos, sin las dudas causadas por la reserva, lo que hizo que ocurrieran cosas extraordinarias. Ocurrieron incluso cosas imposibles. Los hackers del MIT habían descubierto que cuando su deseo de hackear los llevaba a persistir con tanta determinación, las barreras de seguridad, el agotamiento y límites mentales parecían encogerse. Ahora, en el movimiento para acabar con generaciones de control centralizado y antihacker de la industria informática, para cambiar la visión desaprobatoria del mundo sobre las computadoras y la gente de las computadoras, la energía combinada de los hackers de hardware obrando unidos podía lograr cualquier cosa. Si no se contenían, si no se replegaban sobre sí mismos, si no se rendían a la fuerza de la codicia, podrían hacer que los ideales del hackerismo se extendieran por la sociedad como si una perla hubiera caído en una vasija de plata.

El Homebrew Club estaba sentado sobre el poder del vril.

Capítulo 11. Tiny BASIC

Mientras que el hambre por construir y expandir el Altair era tan insaciable en los hackers de hardware de los setenta como el deseo de hackear PDP-1 y 6 lo era para los hackers del MIT de los sesenta, se estaba desarrollando un conflicto alrededor del Homebrew Computer Club que tenía el potencial de frenar el proceso idealista de arranque y detener la marea creciente que los elevaba a todos. En el centro del problema estaba uno de los principios centrales de la Ética Hacker: el libre flujo de información, particularmente información que ayudaba a los hackers a entender, explorar y construir sistemas. Anteriormente, no había habido mucho problema en obtener esa información de otros. El tiempo de la "sección de mapeo" en Homebrew era un buen ejemplo de eso: a menudo se revelaban secretos que las grandes empresas institucionales consideraban propietarios. Y en 1976 había más publicaciones que se sumaban a lo que se estaba convirtiendo en una red nacional de hackers de hardware (además de PCC y el boletín Homebrew, ahora estaba la revista Byte en New Hampshire); siempre se podían encontrar programas interesantes en lenguaje ensamblador, sugerencias de hardware y chismes técnicos. Las nuevas compañías formadas por hackers entregaban esquemas de sus productos en Homebrew, sin preocuparse de si los competidores podrían verlos; y después de las reuniones en The Oasis, los jóvenes ejecutivos de las diferentes compañías, vestidos con jeans azules, discutían libremente cuántas placas entregaban y qué nuevos productos estaban considerando.

Luego vino la protesta por Altair BASIC. Les daría a los hackers de hardware una pista de la nueva fragilidad de la ética hacker. E indicaría que, a medida que el poder de las computadoras llegara a la gente, otras filosofías menos altruistas podrían prevalecer.

Todo comenzó como una típica travesura hacker. Entre los productos que MITS había anunciado, pero que aún no había enviado a quienes lo habían pedido, había una versión del lenguaje informático BASIC. Entre las herramientas que podía tener un propietario de un Altair, ésta era una de las más codiciadas: porque, una vez que tuviera un BASIC en su Altair, la capacidad de la máquina para implementar sistemas, para mover pirámides mentales, mejoraría “en órdenes de magnitud”, como decía la expresión. En lugar de tener que escribir laboriosamente programas en lenguaje de máquina en cintas de papel y luego tener que volver a traducir las señales (para entonces muchos propietarios de Altair habían instalado tarjetas de E/S que les permitían conectar las máquinas a teletipos y lectores de cintas de papel), tendría una forma de escribir programas rápidos y útiles. Mientras que los hackers de software (y ciertamente fanáticos del lenguaje ensamblador tan antiguos como Gosper y Greenblatt) desdeñaban el BASIC por considerarlo un lenguaje fascista, para los hackers de hardware que intentaban ampliar sus sistemas era una herramienta increíblemente valiosa.

El problema era que al principio no se podía conseguir un BASIC. Era particularmente enloquecedor porque MITS supuestamente tenía uno, aunque nadie en Homebrew lo había visto funcionar.

De hecho, MITS tenía un BASIC. El lenguaje ya estaba en uso desde principios de la primavera de 1975. Poco antes de que MITS comenzara a enviar Altairs a los lectores de Popular Electronics, hambrientos de computadoras, Ed Roberts había recibido una llamada telefónica de dos estudiantes universitarios llamados Paul Allen y Bill Gates.

Los dos adolescentes eran de Seattle. Desde la secundaria, los dos habían estado hackeando computadoras; grandes empresas les pagaban por realizar lucrativos contratos de programación.

Cuando Gates, un genio delgado y rubio que parecía incluso más joven que sus tiernos años, se fue a Harvard, los dos descubrieron que se podía ganar algo de dinero fabricando intérpretes para lenguajes de computadora como BASIC para computadoras nuevas.

El artículo de Altair, aunque no los impresionó técnicamente, los entusiasmó: estaba claro que los microordenadores eran la próxima gran novedad y podían participar en toda la acción escribiendo BASIC para esta cosa. Tenían un manual que explicaba el conjunto de instrucciones para el chip 8080 y tenían el artículo de Popular Electronics con los esquemas de Altair, así que se pusieron a trabajar en escribir algo que cupiera en 4K de memoria. En realidad, tuvieron que escribir el intérprete en menos de esa cantidad de código, ya que la memoria no sólo albergaría su programa para interpretar BASIC en lenguaje de máquina, sino que también necesitaría espacio para el programa que el usuario estaría escribiendo. No fue fácil, pero Gates en particular era un maestro en escribir código, y con mucho esfuerzo y un uso innovador del elaborado conjunto de instrucciones del 8080, pensaron que lo habían logrado. Cuando llamaron a Roberts, no mencionaron que estaban haciendo la llamada desde el dormitorio universitario de Bill Gates. Roberts fue cordial, pero les advirtió que otros estaban pensando en un Altair BASIC; sin embargo, eran bienvenidos a intentarlo. "Le compraremos al primer tipo que aparezca con uno", les dijo Roberts.

Poco después, Paul Allen estaba en un avión a Albuquerque con una cinta de papel que contenía lo que él y su amigo esperaban que ejecutara BASIC en la máquina. Encontró que MITS era un manicomio. "La gente trabajaba todo el día, corría a casa, comía la cena y regresaba", recordó más tarde el ejecutivo de MITS, Eddie Currie. "Podrías ir a cualquier hora del día o de la noche y encontrar a veinte o treinta personas, entre un tercio y la mitad del personal [excluyendo la fabricación]. Y esto se hacía siete días a la semana. La gente estaba realmente involucrada en esto porque estaban dando computadoras a personas que estaban muy agradecidas y que las querían tanto. Fue una cruzada grandiosa y gloriosa".

Solo una máquina en MITS tenía 4K de memoria, y apenas funcionaba. Cuando Paul Allen introdujo la cinta en el lector de teletipo y leyó la cinta, nadie estaba seguro de lo que sucedería. Lo que sucedió fue que el teletipo al que estaba conectado dijo: LISTO. ¡Listo para programar! "Se emocionaron mucho", dijo Bill Gates más tarde. "Nadie había visto nunca a la máquina hacer nada".

El BASIC estaba lejos de ser una versión funcional, pero estaba lo suficientemente cerca de completarse y sus rutinas eran lo suficientemente inteligentes como para impresionar a Ed Roberts. Contrató a Allen y se encargó de que Gates trabajara desde Harvard para ayudar a que la cosa funcionara. Cuando, poco después, Gates finalmente dejó la escuela (nunca volvería) para ir a Albuquerque, se sintió como Picasso tropezando con un mar de lienzos en blanco: allí había una computadora ordenada sin utilidades. "¡No tenían nada!", dijo más tarde, asombrado años después del hecho. "Quiero decir, el lugar no era sofisticado, en lo que se refiere al software. Reescribimos el ensamblador, reescribimos el cargador... armamos una biblioteca de software. Era un material bastante cutre, pero la gente podía divertirse usándolo".

La diferencia entre la biblioteca de software de Gates-Allen y la biblioteca de software en el cajón junto al PDP-6 o la biblioteca del Homebrew Club era que la primera estaba a la venta únicamente. Ni Bill Gates ni Ed Roberts creían que el software fuera algún tipo de material santificado, destinado a pasarse de mano en mano como si fuera demasiado sagrado para pagar por él.

Representaba trabajo, al igual que el hardware, y Altair BASIC figuraba en el catálogo de MITS como cualquier otra cosa que vendiera.

Mientras tanto, la sed en el Homebrew Club por un Altair BASIC se estaba volviendo insoportable. Resultó que los miembros de Homebrew eran perfectamente capaces de escribir intérpretes BASIC, y de hecho algunos de ellos lo harían. Otros, sin embargo, habían pedido Altair BASIC y esperaban impacientemente su entrega, tal como habían esperado impacientemente la entrega de otros productos MITS. La paciencia con MITS se estaba agotando, especialmente desde la debacle con las placas de memoria dinámica que Ed Roberts insistía en que debían funcionar y nunca lo hicieron. Aquellos estafadas por comprar placas de memoria comenzaron a resoplar y hacer pucheros cuando hablaban de la compañía de Ed Roberts, especialmente porque el propio Roberts, que había alcanzado un estatus legendario como un genio solitario que nunca salía de Albuquerque, era mencionado como un enemigo codicioso y ávido de poder de la Ética Hacker. Incluso se rumoreaba que deseaba el mal a sus competidores. La respuesta adecuada de un hacker a los competidores era darles su plan de negocios e información técnica, para que pudieran hacer mejores productos y el mundo en general pudiera mejorar. No actuar como lo hizo Ed Roberts en la Primera Convención Mundial Altair, celebrada en Albuquerque un año después de que se introdujeran las máquinas, cuando el tenaz presidente de MITS se negó a alquilar stands de exposición a los competidores y, según algunos, se enfureció cuando se enteró de que empresas como Processor Technology, de Bob Marsh, habían alquilado suites en el hotel de la convención y estaban entreteniendo a posibles clientes.

Por eso, cuando la Caravana de MITS llegó al Rickeys Hyatt House de Palo Alto en junio de 1975, el escenario estaba preparado para lo que algunos llamarían un crimen y otros llamarían liberación. La "Caravana" fue una innovación de marketing de MITS. Algunos de los ingenieros de MITS viajarían en una autocaravana, llamada MITS-mobile, de ciudad en ciudad, instalando Altairs en salas de seminarios de moteles e invitando a la gente a ver los asombrosos ordenadores de bajo coste en funcionamiento. La asistencia estaría formada en gran medida por gente que había pedido Altairs y tenía preguntas sobre cuándo podrían esperar la entrega. La gente que los poseía querría saber en qué se habían equivocado al montar el monstruo. La gente que poseía placas de memoria MITS querría saber por qué no funcionaban, y la gente que había pedido Altair BASIC se quejaba de que no lo habían recibido.

La multitud del Homebrew Computer Club salió en masa cuando la Caravana se reunió en el Rickeys Hyatt de El Camino Real en Palo Alto a principios de junio, y se sorprendieron cuando descubrieron que el Altair que se exhibía funcionaba con BASIC. Estaba conectada a un teletipo que tenía un lector de cinta de papel y, una vez cargado, cualquiera podía escribir comandos y obtener respuestas al instante. Parecía una bendición para aquellos hackers que ya habían enviado varios cientos de dólares a MITS y esperaban con impaciencia el BASIC. No hay nada más frustrante para un hacker que ver una extensión de un sistema y no poder controlarlo. La idea de volver a casa y encontrar un Altair sin la capacidad de esa máquina para funcionar en los confines pseudo-lujosos del Rickeys Hyatt debe haber sido como una sentencia de prisión para esos hackers. Pero la experiencia prevaleció. Años después, Steve Dompier - con mucho tacto - describió lo que sucedió a continuación: “Alguien, no creo que nadie haya descubierto quién, tomó prestada una de sus cintas de papel que estaban en el suelo”. La cinta de papel en cuestión contenía la versión actual del Altair BASIC escrita por Bill Gates y Paul Allen.

Dan Sokol recordó más tarde que ese vago “alguien” se le acercó y, notando que Sokol trabajaba para una de las empresas de semiconductores, le preguntó si tenía alguna forma de duplicar las cintas de papel. Sokol dijo que sí, que había una máquina copiadora de cintas a su disposición. Le entregaron la cinta.

Sokol tenía todo tipo de razones para aceptar la tarea de copiar las cintas. Pensaba que el precio de MITS por el BASIC era excesivo. Pensaba que MITS era codiciosa. Había oído un rumor de que Gates y Allen habían escrito el intérprete en un gran sistema informático perteneciente a una institución financiada en parte por el gobierno y, por lo tanto, consideraba que el programa pertenecía a todos los contribuyentes.

Sabía que mucha gente ya había pagado a MITS por el producto, y que conseguir una copia anticipada no perjudicaría económicamente a MITS. Pero, sobre todo, le parecía correcto copiarlo. ¿Por qué debería haber una barrera de propiedad entre un hacker y una herramienta para explorar, mejorar y construir sistemas?

Armado con este razonamiento filosófico, Sokol llevó la cinta a casa de su jefe, se sentó ante un PDP-11 y enhebró la cinta. Lo dejó funcionando toda la noche, produciendo cintas, y en la siguiente reunión del Homebrew Computer Club llegó con una caja de cintas. Sokol cobraba lo que en términos de hacker era el precio adecuado para el software: nada. La única condición era que si uno se llevaba una cinta, debía hacer copias y acudir a la siguiente reunión con dos cintas. Y regalarlas. La gente se apoderó de las cintas, y no sólo llevó copias a la siguiente reunión, sino que también las envió a otros clubes de informática. De modo que esa primera versión de Altair BASIC estaba en libre circulación incluso antes de su lanzamiento oficial.

Sin embargo, había dos hackers que no estaban nada contentos con esta demostración de colaboración y colaboración: Paul Allen y Bill Gates. Habían vendido su BASIC a MITS sobre la base de que les reportaban regalías por cada copia vendida, y la idea de que la comunidad hacker produjera alegremente copias de su programa y las regalara no parecía particularmente utópica. Parecía un robo. Bill Gates también estaba molesto porque la versión que la gente intercambiaba estaba llena de errores que él estaba en proceso de corregir. Al principio pensó que la gente simplemente compraría la versión depurada. Pero incluso después de que MITS lanzara el BASIC depurado, quedó claro que los usuarios de Altair no compraban tantas copias como lo harían si no hubieran tenido un BASIC “pirateado” ya en funcionamiento. Aparentemente, o bien estaban aguantando los errores o, más probablemente, estaban pasándoselo en grande como hackers depurándolo ellos mismos. Gates se estaba enfadando mucho, y cuando David Bunnell (que por entonces editaba el recién creado boletín de usuarios de Altair para MITS) le preguntó qué quería hacer al respecto, Gates, que por entonces tenía diecinueve años y estaba imbuido de una arrogancia que proviene del virtuosismo técnico y no necesariamente del tacto social, dijo que tal vez debería escribir una carta. Bunnell le prometió que podría hacerla llegar a los alborotadores.

Así que Gates escribió su carta, y Bunnell no sólo la publicó en el boletín de Altair, sino que la envió a otras publicaciones, incluido el boletín del Homebrew Computer Club. En la carta, titulada "Carta abierta a los aficionados", se explicaba que, si bien él y Allen habían recibido muchos comentarios positivos sobre el intérprete, la mayoría de las personas que lo elogiaban no lo habían comprado. La carta llegaba al meollo del asunto rápidamente:

> ¿Por qué sucede esto? Como la mayoría de los aficionados deben saber, la mayoría de ustedes roban su software. El hardware debe pagarse, pero el software es algo para compartir. ¿A quién le importa si la gente que trabajó en él recibe su pago?

Gates continuó explicando que este "robo" de software estaba impidiendo que los programadores talentosos escribieran para máquinas como el Altair. "¿Quién puede permitirse hacer un trabajo profesional a cambio de nada? ¿Quién de los aficionados puede dedicar tres años y medio-hombre a programar, encontrar todos los bugs, documentar su producto y distribuirlo gratuitamente?"

Aunque bastante apasionada, la carta - cuidadosamente editada por Bunnell - estaba lejos de ser una diatriba. Pero desató el infierno en la comunidad hacker. Ed Roberts, aunque estaba de acuerdo filosóficamente con Gates, no pudo evitar notar los malos sentimientos y se sintió molesto porque Gates no lo había consultado antes de publicar la carta. La Southern California Computer Society amenazó con demandar a Gates por llamar a los aficionados "ladrones". Gates recibió entre trescientas y cuatrocientas cartas, de las cuales sólo cinco o seis contenían el pago voluntario que sugirió que le enviaran los propietarios de BASIC pirateado. Muchas de las cartas eran intensamente negativas. Hal Singer, editor del Micro-8 Newsletter, que recibió la carta de Gates por entrega especial, escribió que "la acción más lógica era romper la carta y olvidarse de ella".

Pero el "escándalo del software", como llegó a conocerse, no se podía olvidar fácilmente. Cuando los hackers del MIT escribían software y lo dejaban en un cajón para que otros trabajaran en él, no tenían la tentación de cobrar derechos de autor. Spacewar de Slug Russell, por ejemplo, no tenía mercado (sólo se fabricaron cincuenta PDP-1, y las instituciones que los poseían difícilmente gastarían dinero en comprar un juego espacial). Con el creciente número de ordenadores en uso (no sólo Altairs, sino también otros), un buen software se convirtió en algo que podía generar mucho dinero, si los hackers no consideraban que hackear el software era algo que les correspondía. Nadie parecía oponerse a que un autor de software recibiera algo por su trabajo, pero los hackers tampoco querían abandonar la idea de que los programas informáticos pertenecían a todo el mundo. Era una parte demasiado importante del sueño hacker como para abandonarla.

Steve Dompier pensaba que Bill Gates simplemente se quejaba. "Irónicamente, las quejas de Bill sobre la piratería no detuvieron nada. La gente todavía creía: 'Si lo tienes, puedes ejecutarlo'. Era como grabar música del aire. BASIC se había extendido por todo el país, por todo el mundo. Y ayudó a Gates: el hecho de que todo el mundo tuviera Altair BASIC y supiera cómo funcionaba y cómo solucionarlo significó que cuando otras empresas de ordenadores entraron en escena y necesitaron un BASIC, acudieron a la empresa de Gates. Se convirtió en un estándar de facto".

La gente del Homebrew Computer Club intentó adaptarse a esta nueva era, en la que el software tenía valor comercial, sin perder los ideales de los hackers. Una forma de hacerlo era escribiendo programas con la idea específica de distribuirlos de la manera informal, aunque casi legal, en la que se distribuía Altair BASIC: mediante un esquema de ramificación, de dárselo a los amigos. De modo que el software pudiera seguir siendo un proceso orgánico, con el autor original lanzando el código del programa en un viaje que vería una ronda interminable de mejoras.

• • • • • • • •

El mejor ejemplo de ese proceso orgánico se produjo en la proliferación de intérpretes de "Tiny BASIC". Cuando Bob Albrecht, del PCC, examinó por primera vez su Altair, se dio cuenta inmediatamente de que la única forma de programarlo era con el pesado lenguaje de máquina del

chip 8080. También vio lo limitada que era la memoria. Así que fue a ver a Dennis Allison, un miembro de la junta del PCC que enseñaba informática en Stanford, y le pidió que hiciera algunas notas de diseño para un BASIC simplificado que fuera fácil de usar y no ocupara mucha memoria. Allison escribió un marco para un posible intérprete, etiquetando su artículo como un "proyecto participativo", solicitando ayuda de cualquier otra persona interesada en escribir "un lenguaje minimalista similar al BASIC para escribir programas simples". Allison recordó más tarde la reacción al artículo de PCC: “Tres semanas después recibimos respuestas, incluida una enviada por dos chicos de Texas que habían escrito un Tiny BASIC totalmente corregido y depurado, con un listado completo de código en octal”. El dúo de Texas había colocado un BASIC en 2K de memoria y lo había enviado, así de simple, para que lo imprimieran en PCC. Albrecht cumplió, ejecutó todo el código fuente y en pocas semanas los propietarios de Altair comenzaron a enviar “informes de errores” y sugerencias para mejorar. Esto fue antes de que existieran las placas de E/S para Altair; los lectores de PCC habían estado cambiando los dos mil números a mano, repitiendo el proceso cada vez que encendían la máquina.

Varios hackers inundaron PCC con nuevos dialectos de Tiny BASIC y programas interesantes escritos en el lenguaje. Albrecht, siempre más planificador que hacker, estaba preocupado de que ejecutar todo ese código convirtiera a PCC en una revista demasiado técnica, por lo que ideó un plan para publicar una rama temporal de PCC llamada Tiny BASIC Journal. Pero la respuesta fue tan contundente que se dio cuenta de que hacía falta una revista enteramente nueva, dedicada al software. Llamó a Jim Warren para que la editara.

Warren era el corpulento y voluble estudiante de informática que se negó a ir a The Oasis después de Homebrew porque no soportaba el humo. Era un veterano de la Universidad Libre de Midpeninsula. Además de varios títulos académicos, tenía unos ocho años de experiencia como consultor en informática y era presidente de varios grupos de interés especial de la Association for Computer Machinery. PCC le ofreció 350 dólares al mes por el trabajo y lo aceptó de inmediato. "Parecía divertido", explicó más tarde. Sabiendo que algunas personas se oponían militantemente a BASIC, insistió en que la revista no se limitara a BASIC sino que publicara software en general, para ayudar a todos aquellos hackers de hardware que habían configurado sus máquinas y querían los conjuros para mover los bits dentro de ellas.

El nombre de la revista era indicativo de la atmósfera que rodeaba a PCC y Homebrew en esa época: como Tiny BASIC ahorra bytes de memoria, se la denominó “La revista del Dr. Dobbs sobre calistenia y ortodoncia informáticas... corriendo ligero sin sobrebytes”. ¿Por qué no?

La revista del Dr. Dobbs (DDJ) - como escribió Warren en su editorial en el primer número - trataría sobre “software gratuito y muy económico”. En una carta enviada para explicar la revista, explicó: “Existe una alternativa viable al problema del software alertado por Bill Gates en su iracunda carta a los hobbistas del cómputo “arrancándole su software”. Cuando el software es gratuito, o es tan barato que es más sencillo pagar por el que duplicarlo, entonces no será ‘robado’”.

Warren consideraba que DDJ era el buque insignia del sueño hacker. Quería que fuera un centro de intercambio de ensambladores, depuradores, gráficos y software musical. También lo veía como un

“medio de comunicación y agitación intelectual”. Pero las cosas estaban sucediendo tan rápido en 1976 que - con frecuencia - las noticias sobre hardware que escuchaba o la solución de software a un problema no podían esperar a ser publicadas, y él corría a la siguiente reunión de Homebrew, donde se convirtió en una figura familiar, poniéndose de pie y soltando todas las noticias que habían llegado a su escritorio esa semana.

La activa presión de Warren para que se adoptara un enfoque de dominio público para el software no era el único curso de acción. Tal vez la respuesta hacker más característica a la amenaza de que la comercialización pudiera cambiar el espíritu del hacking vino de un mago del software inflexiblemente independiente llamado Tom Pittman. Pittman no estaba involucrado en ninguno de los grandes proyectos que se estaban desarrollando en ese momento en torno al Homebrew. Era un representante de los hackers de hardware de mediana edad que gravitaban hacia el Homebrew y se enorgullecían de asociarse con la revolución de los microordenadores, pero obtenían tanta satisfacción de las alegrías personales del hacking que mantenían un perfil bajo. Pittman tenía la edad de Lee Felsenstein, e incluso había estado en Berkeley al mismo tiempo, pero no vivía la vida interna audaz de Felsenstein.

Pittman había estado asistiendo fielmente al Homebrew desde la primera reunión, y sin hacer mucho esfuerzo por comunicarse, se hizo conocido como uno de los ingenieros más puros y consumados del club. Era un tipo de complexión delgada, con anteojos gruesas y una sonrisa amplia y parpadeante que indicaba, a pesar de una timidez evidente, que siempre estaría dispuesto a participar en conversaciones sobre hardware. Había construido un sistema informático increíblemente útil basado en el chip Intel 4004, de relativamente bajo consumo, y durante un tiempo mantuvo la lista de correo de Homebrew sobre él. Disfrutaba perversamente de provocar el asombro de la gente cuando les contaba lo que había hecho con el sistema, obligándolo a realizar tareas que iban mucho más allá de sus límites teóricos.

Pittman había soñado con tener su propio ordenador desde que estaba en el instituto, a principios de los años sesenta. Toda su vida se había descrito a sí mismo como un «hacedor, no un observador», pero trabajaba solo, en un mundo privado dominado por la lógica tranquilizadora de la electrónica. «No soy muy sensible a los patrones de pensamiento de los demás», dijo más tarde. Iba a la biblioteca a buscar libros sobre el tema, los hojeaba y luego sacaba más. «No podía leer mucho antes de dejar el libro y hacer cosas, en mi cabeza si no en ningún otro lugar».

Cuando llegó a Berkeley, ya había tomado cursos de nivel universitario sobre todo tipo de materias de matemáticas e ingeniería. Su asignatura favorita durante su primer año fue Análisis Numérico. Mientras el Movimiento por la Libertad de Expresión hacía estragos a su alrededor, Pittman se enredaba alegremente con los problemas de la sección de laboratorio del curso, luchando sistemáticamente con cada enigma matemático hasta que aullaba pidiendo clemencia. Pero la parte teórica del curso le aburría; no parecía “interesante”, y su nota en Análisis Numérico se dividió entre una A en el laboratorio y una F en la clase. Obtuvo resultados idénticos al repetir el curso. Tal vez no estaba destinado a encajar en la estructura organizada de una universidad.

Entonces encontró su escape. Un profesor comprensivo lo ayudó a conseguir un trabajo en un laboratorio del Departamento de Defensa en San Francisco. Allí trabajó con computadoras, ayudando en simulaciones de juegos que medían el efecto de la radiación de explosiones nucleares hipotéticas. No tenía ningún problema ético con el trabajo. “Como soy básicamente insensible a las cuestiones políticas, ni siquiera me di cuenta”, dijo más tarde. Sus creencias como cristiano devoto

lo llevaron a declararse un “semi-objetor”. Más tarde explicó: “Significa que estaba dispuesto [a servir] pero no a disparar a la gente. Trabajé allí en el laboratorio para servir a mi país. Me divertí mucho”.

Agradeció la oportunidad de finalmente volverse adicto a las computadoras; aunque su horario de trabajo terminaba oficialmente a las seis, a menudo trabajaba mucho más tarde, disfrutando de la paz de ser el único allí. Trabajaba hasta que estaba demasiado cansado para continuar; una noche, mientras conducía de regreso a casa en East Bay, se quedó dormido y se despertó en un rosal al costado de la carretera. Aprendió el sistema informático en el laboratorio tan bien que se convirtió en el hacker de sistemas no oficial; siempre que la gente tenía un problema con la máquina, acudían a Tom. Se sintió destrozado cuando, después de que la guerra terminó y los fondos de defensa se agotaron, el laboratorio cerró.

Pero para entonces, la posibilidad de fabricar su propia computadora se había materializado. Fue a Intel, fabricante del primer microprocesador, el chip 4004, y se ofreció a escribir un ensamblador para él. Tomaría las piezas para construir una computadora a cambio del trabajo. Tom Pittman, que Descifrando código como un maestro, creó un ensamblador compacto y luego escribió un depurador a cambio de más piezas. La gente de Intel empezó a enviar a Tom a todos los compradores de 4004 que necesitaban programación. Cuando empezó a asistir a las reuniones de Homebrew, ya se había mudado a San José, tras haber creado un importante negocio de consultoría para mantenerse a sí mismo y a su esposa, que aceptaba su fanatismo informático sólo a regañadientes.

Aunque le fascinaba la hermandad tecnológica del Homebrew, Tom Pittman estaba entre aquellos que nunca consideraron entrar en el mundo de los negocios como Bob Marsh hizo con Processor Technology. Tampoco pensó en trabajar en ninguna de esas enérgicas empresas emergentes. “Nunca me llevé bien con nadie allí. La gente no me conocía; soy un solitario”, dijo más tarde. “Además, no tengo habilidades de gestión. Soy más una persona de software que un ingeniero electrónico”.

Pero después del “escándalo del software” provocado por la carta de Bill Gates, Pittman decidió emprender acciones públicas. “Gates se quejaba de las estafas y la gente decía: ‘Si no cobraras 150 dólares, lo compraríamos’. Decidí demostrarlo”. Había estado siguiendo las noticias sobre Tiny BASIC en el Dr. Dobbs Journal y entendía las pautas para escribir un BASIC. Y notó que estaban apareciendo algunos nuevos ordenadores, competidores de MITS, que utilizaban el chip Motorola 6800 en lugar del Intel 8080, y no había ningún intérprete BASIC escrito para que funcionara en ellos. Así que decidió escribir un intérprete Tiny BASIC 6800 y venderlo por la suma de cinco dólares, una fracción del precio de MITS, para ver si la gente lo compraría en lugar de robar.

Como era un verdadero hacker, Pittman no se conformaba con ejecutar cualquier tipo de Tiny BASIC: era prisionero de la bestia a la que llamaba “la criatura espeluznante de las características”, que se coloca detrás del hombro de cada hacker, dándole golpecitos en la espalda y urgiéndole: “¡Más características! ¡Hazlo mejor!”. Incluyó cosas que algunas personas consideraban imposibles en un lenguaje “pequeño”, como espacio para insertar comentarios útiles y la utilización de un conjunto completo de comandos. En dos meses ya tenía su intérprete funcionando y tuvo suerte cuando lo vendió a la compañía AMI por 3.500 dólares, con la condición de que la venta no fuera exclusiva. Aun así quería venderlo a aficionados por cinco dólares cada uno.

Envió un anuncio a la revista Byte y, a los pocos días de su aparición, tenía cincuenta dólares en su buzón. Algunas personas enviaron diez dólares o más, diciendo que los cinco dólares eran demasiado poco. Algunos enviaron cinco dólares con una nota diciendo que no les enviaran nada,

ya que lo habían copiado de un amigo. Pittman siguió enviándolos. Sus costos incluían doce centavos por la cinta de papel y cincuenta centavos por la impresión del manual que había escrito. Se sentaba en el sofá de su modesta casa por la noche, escuchando la estación de radio cristiana de San José o grabando casetes de oradores en conferencias cristianas y doblando cintas de papel, habiendo dominado la técnica de doblar cada veinte centímetros. Después iba a la oficina de correos y enviaba los paquetes. Todo lo hacía a mano, con la ayuda de su mujer, que se había mostrado escéptica con todo el asunto.

Fue un triunfo para el hackerismo, pero Tom Pittman no se detuvo allí. Quería contárselo a la gente, mostrarles el ejemplo con el que podían crecer. Más tarde hizo una presentación en una reunión de Homebrew y, cuando se dirigió a paso rápido al frente del auditorio, Lee vio que tenía el cuerpo tenso. Lee intentó relajarlo: "Te llaman el pequeño Tom Pittman, pero en realidad no eres tan pequeño", dijo. "¿Por qué?". Tom, que no estaba acostumbrado a las réplicas públicas, no respondió con más que una risa. Pero cuando empezó a hablar, cobró fuerza, encogiendo y desenrollando su cuerpo, agitando el brazo en el aire para hacer comentarios sobre el software libre. Tenía cierto dramatismo, este técnico normalmente taciturno hablando con sincera franqueza sobre un tema que obviamente le importaba: el libre flujo de información.

Poco después de Tiny BASIC, dio un paso más y anunció su intención de escribir un FORTRAN para microcomputadoras y venderlo por veinticinco dólares. Se trataba de otra empresa entusiasta a tiempo completo, y todavía estaba trabajando en ello cuando, como dijo más tarde, "mi viuda de la computadora me dejó. Decidió que no quería casarse con un adicto".

Fue un sobresalto que experimentarían muchos miembros del Homebrew, aquellos que habían convencido a una mujer para que se casara con un adicto a la computadora en primer lugar. "Yo diría que la tasa de divorcios entre los informáticos es casi del cien por ciento, ciertamente en mi caso", dijo más tarde Gordon French. Eso no le facilitó las cosas a Tom Pittman. No tenía ánimos para terminar el FORTRAN. Pensó mucho en la devoción que le había dado a la máquina y de dónde venía, y se sentó a escribir algo, no en lenguaje de máquina, sino en inglés.

Tituló el ensayo "Deus Ex Machina, o El Verdadero Computerista" (se podría usar la última palabra indistintamente con "hacker"), y fue una explicación reveladora de lo que unía a los hackers de hardware de Silicon Valley y a los hackers de inteligencia artificial de Cambridge. Escribió sobre la sensación que se tiene después de hackear algo. "En ese instante", escribió, "yo, como cristiano, pensé que podía sentir algo de la satisfacción que Dios debe haber sentido cuando creó el mundo". Luego recopiló el credo del informático, el hacker de hardware, que incluía "artículos de fe" tan familiares (para la gente del Homebrew) como:

> El ordenador es más interesante que la mayoría de la gente. Me encanta pasar tiempo con mi ordenador. Es divertido escribir programas para él, jugar en él y construir nuevas partes para él. Es fascinante tratar de averiguar en qué parte del programa está por la forma en que parpadean las luces o zumba la radio. Es mejor que una conversación aburrida cualquier día.
>
> El ordenador necesita sólo un poco más (memoria) (velocidad) (periféricos) (mejor BASIC) (CPU más nueva) (supresión de ruido en el bus) (depuración en este programa) (editor potente) (fuente de alimentación más grande) antes de que pueda hacer esto o aquello.

> No hay necesidad de comprar este paquete de software o esa placa de circuitos; puedo diseñar uno mejor.
>
> No te pierdas nunca una reunión del club. Aquí es donde está la cosa. Las pequeñas noticias jugosas, las instrucciones para solucionar el problema que me ha estado molestando las últimas dos semanas... ¡eso es lo real! Además, puede que tengan algún software gratuito.

El tono de Pittman cambió en ese punto. Se obligó a sí mismo a hacer una excepción a esos artículos de fe, testificando que había "estado allí" y había visto los problemas que tenían. Punto por punto demostró la locura de hackear y concluyó escribiendo: "A estas alturas, el ordenador ha salido del estudio y se ha instalado en el resto de tu vida. Consumirá todo tu tiempo libre, e incluso tus vacaciones, si se lo permites. Vaciará tu billetera y atará tus pensamientos. Ahuyentará a tu familia. Tus amigos empezarán a pensar que eres un aburrido. ¿Y para qué?". Conmocionado por la ruptura de su matrimonio, Tom Pittman decidió cambiar sus hábitos. Y lo hizo. Más tarde describió la transformación: “Ahora me tomo un día de descanso. No encenderé la computadora el domingo”.

“Los otros seis días trabajaré como un perro”.

• • • • • • • •

Lee Felsenstein estaba ganando confianza y propósito a través de su papel como maestro de ceremonias del Homebrew Computer Club. Su deseo expreso era permitir que el club se desarrollara como una comunidad anarquista, una sociedad de no afiliados casados, lo supieran o no, por la Propaganda de la Escritura. Vio lo que Moore y French no vieron: para obtener el máximo efecto político en la guerra de los hackers de hardware contra las fuerzas malignas de IBM y demás, la estrategia debería reflejar el estilo del hackerismo en sí. Esto significaba que el club nunca sería manejado como una burocracia formal.

Si quería un modelo para el fracaso, sólo tenía que mirar hacia el sur, a la Southern California Computer Society. La SCCS, que comenzó a funcionar unos meses después de la primera reunión de Homebrew, aprovechó a los aficionados del sector de la electrónica intensiva (casi todos los contratistas de defensa de alta tecnología están en el sur de California) para aumentar rápidamente su membresía a ocho mil. Sus líderes no estaban contentos con el mero intercambio de información: imaginaban planes de compra en grupo, una revista nacional y una influencia que permitiría a los aficionados dictar las condiciones a la creciente industria de la microcomputadora. Homebrew no tenía un comité directivo que deliberara sobre objetivos y direcciones; sólo se constituyó como una idea de último momento, casi un año después de su creación; no tenía requisitos reales de cuotas: sólo una contribución sugerida de diez dólares al año para obtener su modesto boletín. Pero la SCCS tenía una junta directiva formal, cuyas reuniones regulares a menudo se desencadenaban por acalorados debates sobre lo que debería ser el club. No pasó mucho tiempo hasta que la SCCS empezó a publicar una elegante revista, a tener un programa de compras en grupo en crecimiento (hasta cuarenta mil dólares al mes) y a considerar cambiar su nombre a National Computer Society.

Bob Marsh, que promocionaba las juntas directivas de Processor Technology, volaba a menudo a las abarrotadas reuniones de la SCCS e incluso formó parte de la junta directiva durante unos meses. Más tarde describió la diferencia entre los dos grupos: “Homebrew era un lugar donde la gente se reunía misteriosamente, dos veces al mes. Nunca fue una organización. Pero la SCCS estaba más organizada. Esos tipos tenían megalomanía. La política era terrible y la arruinó”. De alguna manera,

los detalles nunca se aclararon; se perdió mucho dinero en el plan de compras. El editor que contrataron para dirigir la elegante revista se sintió justificado al abandonar la relación de la publicación con el club y emprender su propio camino con la revista (que seguía publicando como Interface Age); el resultado fue una demanda. Las reuniones de la junta directiva se volvieron increíblemente tempestuosas y los malos sentimientos se extendieron a las reuniones generales de los miembros. Finalmente, el club se desvaneció.

Aunque los planes de Lee no eran menos ambiciosos que los de los líderes de la SCCS, se dio cuenta de que esta guerra no debía librarse de forma burocrática, siguiendo al líder. Estaba perfectamente contento tratando con un ejército de Bob Marshes y Tom Pittmans, algunos de los cuales cambiaron el mundo gracias a productos útiles fabricados en el espíritu del hackerismo y otros que simplemente siguen su camino, siendo hackers. El objetivo final sería una distribución masiva del asombro que Lee Felsenstein había experimentado en su monasterio del sótano. Un entorno propicio para el Imperativo de la Manos a la Obra. Como dijo Lee en una conferencia del Instituto de Ingenieros Eléctricos y Electrónicos en 1975, "El enfoque industrial es sombrío y no funciona: el lema del diseño es 'Diseñado por genios para su uso por idiotas', y la consigna para tratar con el público inexperto y descuidado es ¡NO TOCAR!... El enfoque cordial que sugiero se basaría en la capacidad del usuario para aprender y obtener cierto control sobre la herramienta. El usuario tendrá que pasar una cierta cantidad de tiempo explorando el interior del equipo, y tendremos que hacer que esto sea posible y no fatal, ni para el equipo ni para la persona".

El equipo al que se refería Felsenstein era su Terminal Tom Swift, que todavía no se había construido en 1975. Pero se estaba acercando. Bob Marsh, ansioso por ampliar el alcance de su floreciente empresa Processor Technology, le ofreció a Lee un trato que no pudo rechazar. "Te pagaré por diseñar la parte de video de la Terminal Tom Swift", le dijo. A Lee le pareció bien, ya que había estado trabajando en documentación y esquemas para Processor Technology desde el principio. Bob Marsh, en el primer año de actividad de la empresa, se adhirió a la ética hacker. La empresa distribuía esquemas y código fuente para software, gratis o a un costo nominal. (En reacción parcial al elevado precio del BASIC del MIT, Processor Technology desarrollaría el suyo propio y lo vendería, junto con el código fuente, por cinco dólares). Durante un tiempo, la empresa tuvo una estructura salarial socialista de 800 dólares al mes para todos los empleados. "No prestábamos atención a las ganancias ni a la gestión de casi ningún tipo".

Lee no era un empleado, y optó por trabajar bajo contrato. "Les daba un precio", recordó Lee más tarde, "y ellos tenían que aumentar el precio por un factor de diez, ya que yo era un pensador de poca monta. En términos de dinero".

En menos de tres meses, Lee había hecho un prototipo funcional. El "módulo de visualización de vídeo" (VDM) de Lee encarnaba una filosofía diferente a la de la otra placa de vídeo para Altair, Dazzler de Cromemco. El Dazzler utilizaba colores y producía sus llamativos efectos volviendo constantemente a la memoria del chip principal del Altair (o de cualquiera de los otros ordenadores nuevos que utilizaban un bus de hardware similar). A Steve Dompier le gustaba utilizar su Dazzler mientras ejecutaba BASIC: arrojaba patrones en la pantalla que daban una impresión visual similar a la de Rorschach de la memoria del ordenador en un momento dado, una salida críptica que daba pistas sobre el funcionamiento del programa, muy parecida a la impresión auditiva que daba de la memoria del TX-0 el altavoz situado debajo de la consola.

Sin embargo, el módulo de visualización de vídeo de Lee era un equipo más centrado en lo estridente, diseñado teniendo en mente la futura reforma de la Memoria Comunitaria. Su salida era en blanco y negro, y en lugar de utilizar puntos, formaba caracteres alfanuméricos. (Lee consideró añadir otra alternativa, hexagramas, como los que se encuentran en el I Ching, pero esa idea quedó archivada de alguna manera). Sin embargo, lo más inteligente del VDM de Lee era la forma en que utilizaba la velocidad de los nuevos chips de microprocesador para permitir que la memoria de la máquina se compartiera entre las tareas de cálculo y las de visualización. Funcionaba como un minisistema de tiempo compartido, donde los dos usuarios eran la pantalla de vídeo y el propio ordenador. El VDM, junto con un Altair y otras tarjetas de expansión, hizo realidad la promesa de la TV Typewriter y fue un éxito instantáneo, aunque, como casi todos los productos de Processor Technology, no estuvo listo hasta poco después de la fecha de entrega prometida, a finales de 1975.

Una persona que quedó particularmente impresionada por el VDM fue Les Solomon en Nueva York. No se conformó con regodearse en la gloria reflejada del lanzamiento de la máquina seminal de Ed Roberts. Su revista había seguido el golpe, había publicado más artículos de portada relacionados con la informática y ahora esperaba presentar una terminal de visualización de vídeo de ordenador completa, un artículo autónomo que tendría la potencia del ordenador además de capacidad de visualización. Sería el siguiente paso más allá del Altair, una combinación de ordenador-teletipo con vídeo. Se acabaron los malditos dedos ensangrentados por los interruptores del panel frontal del Altair. En busca del producto, Solomon fue a Phoenix para visitar a Don Lancaster, inventor de la máquina de escribir televisiva (la que Bob Marsh había intentado construir en Berkeley), y lo convenció de que fuera a Albuquerque para reunirse con Ed Roberts, tal vez los dos gigantes podrían combinarse en un proyecto de terminal. Como Solomon lo describió más tarde, la reunión fue "bang, choque. Un choque de egos. Don se negó a cambiar su diseño para que coincidiera con el ordenador de Ed porque dijo que el de Ed era ineficiente. Ed dijo: “De ninguna manera, no puedo rediseñarlo”. Inmediatamente decidieron matarse entre ellos en el acto y los separé”.

Entonces Solomon fue a Bob Marsh, cuya compañía Processor Technology ya ofrecía el VDM y la placa de memoria e incluso una “placa madre” que pudiera reemplazar los circuitos básicos del Altair, y preguntó: “¿Por qué no los pones todos juntos? Hagamos algo de lo que podamos estar orgullosos”. Si Marsh podía entregar un “terminal inteligente” en treinta días, Solomon lo pondría en la portada.

Bob habló con Lee, quien aceptó hacer la mayor parte del diseño, y mientras lo discutían se dieron cuenta de que lo que Les Solomon quería no era simplemente un terminal, sino un ordenador completo. En el año transcurrido desde que se anunció el Altair, habían aparecido ordenadores “para aficionados”, que se vendían en kits o ensamblados, en particular uno llamado IMSAI, fabricado por una empresa cuyos empleados habían recibido la formación de Werner Erhard. Casi todos estos ordenadores utilizaban el bus Altair de 100 pines como base. Casi todos se parecían al Altair, un receptor estéreo de gran tamaño con luces e interruptores en lugar de un dial FM. Todos requerían algún tipo de terminal, normalmente un teletipo tosco, para que el usuario pudiera hacer algo con él.

Ese mes, diciembre de 1975, Lee y Bob trabajaron en el diseño. Marsh quería utilizar un chip 8080, una idea a la que Lee al principio todavía se oponía por razones políticas (¿por qué un dictador de silicio centralizado?) pero que acabó aceptando cuando se dio cuenta de que una terminal verdaderamente “inteligente” (una que te diera todo el poder de una computadora) necesitaría un cerebro. Lee pensó que usaría su estilo paranoico de desguace para equilibrar el resto del diseño, de

modo que el cerebro no se sintiera tentado a volverse loco. Marsh interrumpía a menudo el diseño en progreso de Lee para revelar su última inspiración de la “criatura de características”.

Lee relató más tarde este proceso en un artículo de revista: “Cuando [Marsh] tenía poco más de qué preocuparse, continuamente aparecía con nuevas características y economías que de repente quería incorporar al diseño. Explicaba el problema o la oportunidad y luego precedía su solución técnica con un inevitable: ‘Todo lo que tienes que hacer es...’ .’ Si el diseñador fuera una prima donna, la relación terminaría después del segundo incidente de ese tipo, y el diseñador se enojaría por su ‘profesionalismo’ y su ‘interferencia’. Por supuesto, como mi taller estaba en la misma habitación que el suyo, no podría haber llegado muy lejos si hubiera querido salir corriendo furioso”.

Marsh, como Lee, pensaba en la máquina como una herramienta política, así como un producto bueno y divertido para diseñar. “Queríamos hacer que la microcomputadora fuera accesible para los seres humanos”, dijo más tarde. “El público aún no lo sabía, pero la computadora era la cosa que vendría y cada hogar tendría una y la gente podría usar las computadoras para cosas útiles. Realmente no estábamos seguros de qué eran [pero] sentíamos que estábamos participando en un movimiento, de alguna manera”.

Lee sugirió que, dado que estaban poniendo la sabiduría de Salomón en la máquina, debería llamarse Sol. (Les Solomon comentó más tarde: “Si funciona, dirán que Sol es el significado de ‘sun’ en español. Si no funciona, le echarán la culpa a los judíos”). Completar la Sol fue un proceso que llevó seis semanas de jornadas de catorce a diecisiete horas, siete días a la semana. Lee, que vivía a base de zumo de naranja, pasaba horas interminables mirando los espaguetis de Mylar del diseño sobre la mesa de luz fluorescente. Mientras tanto, uno de los amigos carpinteros de Bob Marsh había conseguido una ganga en piezas de nogal cortadas al centro, y se decidió que los lados de la Sol estarían hechos de tan elegante material. Las placas del prototipo finalmente se terminaron, solo quince días después de la fecha límite original de Les Solomon. Dos semanas después - un día antes de la nueva fecha de entrega programada para fines de febrero de 1976 en Nueva York - se apresuraban para que todos los mecanismos encajaran en un bus estilo Altair, junto con una fuente de alimentación improvisada, un teclado e incluso un software preliminar. El sistema operativo fue escrito por el jefe de desarrollo de software de Processor Tech, el computerista casero Steve Dompier.

Marsh, siempre ahorrativo, había reservado un vuelo nocturno para él y Lee. Terminaron justo a tiempo y tuvieron que correr a un helipuerto para tomar el avión. Llegaron a Kennedy alrededor de las 6 a.m., agotados, con la Computadora del Hombre Común distribuida en dos bolsas de papel. No había nada abierto en el aeropuerto, ni siquiera para tomar café, así que Solomon los invitó a su casa en Flushing para desayunar. Para entonces, la casa de Les Solomon, en particular su taller en el sótano, estaba alcanzando un estatus legendario como campo de pruebas para nuevos y emocionantes avances. A menudo entretenía a los jóvenes hackers de hardware que diseñaban estos productos, y su esposa siempre los reconocía de un vistazo. "Porque todos tenían lo mismo", explicaría Solomon más tarde. "Ese pequeño ardor dentro del globo ocular. Ella solía decir que tenían una personalidad interna, y aunque parecían vagos mal entretenidos, los mirabas a la cara, los mirabas a los ojos y sabías quiénes eran. Ella los miraba y lo que salía a la luz era el brillo, la intensidad”.

El brillo se atenuó en esa fría mañana de febrero: la terminal de Marsh y Felsenstein no funcionaba. Después de una rápida excursión de un día a New Hampshire para reunirse con la gente de la nueva

revista hobbista Byte, Lee pudo llegar a un banco de trabajo y encontrar el problema: se había soltado un pequeño cable. Volvieron a las oficinas de Popular Electronics y lo encendieron. “Parecía una casa en llamas”, dijo Solomon más tarde. Inmediatamente se dio cuenta de que estaba viendo un ordenador completo.

El artículo de Popular Electronics resultante hablaba de una terminal de ordenador inteligente. Pero era claramente un ordenador, un ordenador que, cuando Processor Technology lo empaquetó en su bonita caja azul con laterales de nogal, parecía más bien una elegante máquina de escribir sin platina. Había nuevos esquemas para el kit revisado (menos de mil dólares), que, por supuesto, se proporcionaban a cualquiera que quisiera ver cómo funcionaba. Marsh calculó más tarde que recibieron entre treinta y cuarenta mil solicitudes de esquemas. Los pedidos del kit seguían llegando. Parecía que la Sol sería la máquina que sacaría el ordenador del mercado de aficionados y llevaría el hacking a los hogares.

La primera exhibición pública de la Sol fue en una exposición en Atlantic City llamada PC '76. Fue una ocasión extraña, la primera vez que los comerciantes de este negocio de aficionados a la informática se reunieron para mostrar sus productos colectivos. El lugar fue el Hotel Shelbourne, y en aquellos días anteriores a los juegos de azar, la gloria del hotel estaba visiblemente desvanecida. Había agujeros en las paredes, algunas de las puertas de las habitaciones no tenían perillas, el aire acondicionado no funcionaba. Algunos jubilados mayores indignados que vivían en el hotel casi atacaron a Steve Dompier en el ascensor cuando vieron su pelo largo. Aun así, fue una experiencia emocionante. Asistieron casi cinco mil personas, muchas de ellas viajando desde otras partes del país (SCCS organizó una gran excursión en grupo de la que aprovecharon muchas personas del Área de la Bahía). Las empresas inspiradas en el cómputo casero como Processor Tech y Cromemco finalmente conocieron almas similares de otras partes del país, y todos se quedaron despiertos hasta bien entrada la noche, intercambiando consejos técnicos y planeando el futuro.

La Sol recibió mucha atención. Todos los hackers parecían estar de acuerdo en que, con su perfil bajo, su teclado integrado de tipo máquina de escribir y su pantalla de vídeo, la Sol era el siguiente paso. Poco después, Processor Tech consiguió que una Sol apareciera en televisión, en el programa “Tomorrow” de Tom Snyder. Esta personalidad televisiva - normalmente agresiva - se encontró cara a cara con la más reciente manifestación del sueño hacker: un ordenador Sol que ejecutaba un programa de juego escrito por Steve Dompier. El juego se llamaba Target y consistía en un pequeño cañón en la parte inferior de la pantalla con el que el usuario podía derribar una serie de naves espaciales alienígenas, hechas de caracteres alfanuméricos, que navegaban por la parte superior de la pantalla. Era un hack ingenioso y Steve Dompier, como dijo más tarde, “básicamente lo regalé”. Después de todo, el objetivo de escribir esos juegos era ver a la gente divertirse con la máquina.

Target era perfecto para mostrar a Tom Snyder y a la audiencia televisiva una nueva forma de ver a esos monstruos envueltos en maldad, los ordenadores. Imagínense a esos post-hippies sucios capaces de llevar un ordenador a un estudio de televisión, instalarlo y hacer que un analfabeto técnico como Tom Snyder hiciera algo con él. Tom se unió a ellos y, antes de que pudieran decir "corte publicitario", ya estaba muy involucrado (y no bromeaba en absoluto) en derribar alienígenas, que irían recorriendo la pantalla en mayor número a medida que avanzaba el juego, e incluso despacharían pequeños paracaidistas cargados con granadas. Era un desafío que uno se sentía obligado a aceptar. Mientras derribaba a los alienígenas, Tom Snyder notaba que había una sensación de... poder. Una sensación que le daba una pequeña muestra de lo que debía ser utilizar esa máquina para crear realmente. ¿Qué misterios había dentro de esa máquina con forma de

máquina de escribir? Incluso algo tan simple como Target podía hacer que alguien pensara en eso. "Nadie le ha dado una definición todavía", dijo más tarde Steve Dompier, "pero creo que hay algo de magia ahí". En cualquier caso, como recordó más tarde Dompier, “tuvieron que sacar a Tom Snyder del ordenador para que terminara el show”.

Capítulo 12. Woz

Steve Wozniak no se sentaba cerca del frente del auditorio de SLAC junto con Lee Felsenstein durante las reuniones de Homebrew. Su participación en las sesiones de mapeo era poco frecuente. No tenía un gran plan social, no incubaba planes para un asalto al estilo de Community Memory sobre los cimientos de la sociedad procesada por lotes. Reunión tras reunión, Steve Wozniak estaría en la parte de atrás de la sala, junto con un contingente suelto de seguidores de sus hazañas digitales, en su mayoría fanáticos de las computadoras en edad de escuela secundaria atraídos por el gran carisma de su hackeo. Parecía un vagabundo. Su cabello caía al azar sobre sus hombros, tenía el tipo de barba que se dejaba crecer más para obviar el lento acto de afeitarse que para mejorar la apariencia, y su ropa (vaqueros y camisetas deportivas, con poca variación) nunca parecía quedarle del todo bien.

Sin embargo, fue Steve Wozniak, conocido por sus amigos como “Woz”, quien mejor ejemplificaría el espíritu y la sinergia del Homebrew Computer Club. Fue Wozniak y la computadora que él diseñaría lo que llevaría la Ética Hacker, al menos en términos de hackeo de hardware, a su apogeo. Sería el legado del Homebrew.

Stephen Wozniak no llegó a sus opiniones sobre el hackerismo a través de la lucha personal y la reflexión política como lo hizo Lee Felsenstein. Era más como Richard Greenblatt y Stew Nelson: un hacker nato. Creció en Cupertino, California, entre las calles sinuosas bordeadas de pequeñas casas unifamiliares y los edificios de una sola planta con pocas ventanas que sembraron la cosecha de silicio que sería tan central para su existencia. Incluso en la escuela primaria, Wozniak podía quedar tan absorto en reflexiones matemáticas que su madre tenía que golpearle la cabeza para traerlo de vuelta al mundo real. Ganó un concurso de ciencias a los trece años por construir una máquina parecida a un ordenador que podía sumar y restar. Su amigo Alan Baum lo recordó más tarde en Homestead High School: “Vi a un tipo garabateando unos diagramas muy bonitos en un trozo de papel. Le pregunté: ‘¿Qué es eso?’. Me respondió: ‘Estoy diseñando un ordenador’. Había aprendido a hacerlo por sí solo”.

Baum quedó lo suficientemente impresionado como para unirse a este inusual compañero de clase en la búsqueda de acceso a un ordenador, y a través de contactos en el Silicon Valley rico en ingeniería lograron acceder a varios ordenadores de tiempo compartido. Todos los miércoles salían de la escuela y un amigo los llevaba a escondidas a una sala de ordenadores de la empresa Sylvania. Programaban la máquina para hacer cosas como imprimir todas las potencias de dos y encontrar los números primos. Los dos seguían la industria informática con la seria pasión con la que los fanáticos del deporte pueden seguir a sus equipos favoritos. Cada vez que se enteraban de que se lanzaba un nuevo miniordenador, escribían al fabricante, ya fuera Digital o Control Data o quien fuera, y solicitaban el manual, una solicitud que a menudo se satisfacía de forma rutinaria. Cuando llegaba el manual, lo devoraban. Pasaban inmediatamente a la parte que describía el conjunto de instrucciones del ordenador. Observaban cuántos registros tenía la máquina, cómo sumaba, cómo hacía la multiplicación y la división. Podían discernir a partir del conjunto de instrucciones el carácter de la máquina, lo fácil que sería de usar. ¿Era ésta una máquina con la que fantasear? Si lo era, Woz recordó más tarde, “pasaría horas en clase escribiendo código sin poder probarlo jamás”. Una vez, después de recibir un manual para un ordenador Data General Nova, él y Baum se encargaron de rediseñarlo, incluso enviaron su nuevo diseño a la empresa, en caso de que Data General quisiera implementar las sugerencias de dos chicos de secundaria.

“Simplemente nos pareció genial [diseñar ordenadores]”, recordó Baum más tarde. “Parecía algo importante que hacer. El glamour nos atraía. Era divertido”. A medida que avanzaba en la escuela secundaria y Wozniak conseguía más tiempo en los ordenadores para perfeccionar sus habilidades, Baum a menudo se asombraba de los trucos de programación que se le ocurrían a Woz. “Parece que él mismo inventó todos los trucos”, dijo Baum más tarde. “Steve ve las cosas de otra manera”.

Dice: ‘¿Por qué no pruebo esto?’ Está decidido a utilizar todas las técnicas de resolución de problemas que puede porque el diseño ordinario no es lo suficientemente bueno. Tiene que ser el mejor. Hará cosas que nadie haya pensado, utilizará todos los trucos. A veces, utilizando todos los trucos, encuentras mejores formas de hacer las cosas”.

Woz se graduó de la escuela secundaria antes que Baum y se fue a la universidad. Pero unos años más tarde, ambos terminaron trabajando en la misma empresa, la firma de computadoras Hewlett-Packard. Una operación de tecnología extremadamente avanzada, dedicada a computadoras de alto rendimiento que eran como autos Mercedes comparados con los torpes Caddies de IBM, esta era realmente la liga mayor, y Woz era muy feliz allí. Estaba casado, pero las computadoras seguían siendo su prioridad número uno. Además de su trabajo en HP diseñando lógica aritmética para chips de calculadora, también hizo algunos trabajos de diseño adicionales para la compañía de juegos Atari, donde trabajaba otro amigo de la escuela secundaria, Steve Jobs. Esto le proporcionó beneficios colaterales, como la vez que fue a un salón de bolos y se encontró con un videojuego que funcionaba con monedas y que tenía un cartel que prometía una pizza a quienquiera que lograra alcanzar cierto nivel. Luego de ganar una buena cantidad de pizzas, su sorprendido acompañante le preguntó cómo había vencido tan fácilmente al juego. “Yo lo diseñé”, dijo Wozniak entre espasmos de risas.

Woz, un bromista con un inquietante y a veces inmaduro sentido del humor, dirigía un servicio gratuito de llamadas telefónicas desde su casa - “dial-a-joke” - en el que se repartían una cantidad aparentemente interminable de chistes de polacos. No era la única diversión que obtenía del teléfono. Jobs y él se inspiraron después de leer un artículo de 1971 en Esquire sobre un personaje legendario conocido como el Capitán Crunch, que era un devoto constructor de cajas azules, unos dispositivos que permitían hacer llamadas de larga distancia gratis. Jobs y Woz construyeron las suyas propias, y no sólo las utilizaban para hacer llamadas gratis, sino que en un momento dado las vendían de puerta en puerta en los dormitorios de Berkeley. Woz utilizó una vez su caja para ver si podía llamar al Papa; Se hizo pasar por Henry Kissinger y casi llegó a Su Eminencia antes de que alguien del Vaticano se diera cuenta.

Woz llevaba una vida desenfadada, centrada en hackear para HP, hackear por su cuenta y jugar. Le encantaba jugar a juegos, especialmente a los electrónicos como el Pong. También jugaba al tenis; como Bill Gosper jugando al ping-pong, a Wozniak le encantaba darle efecto a la pelota. Como le dijo más tarde a un entrevistador, "ganar no es tan importante como correr detrás de la pelota". Un sentimiento que se aplicaba tanto al hackeo de ordenadores como al tenis.

Siempre soñó con ese ordenador que podría diseñar para sí mismo. Ya había fabricado en casa su propia máquina de escribir televisiva, un buen primer paso. Su objetivo era, por supuesto, un ordenador diseñado para fomentar el hackeo: una herramienta para hacer herramientas, un sistema para crear sistemas. Sería más inteligente que cualquiera de los anteriores.

Era 1975 y la mayoría de la gente, si hubiera oído su sueño, habría pensado que estaba loco.

Entonces Alan Baum vio el anuncio de la reunión de Homebrew en un tablón de anuncios y se lo contó a Woz. Ambos fueron. Baum, que admitió ser demasiado perezoso para construir un ordenador cuando estaba rodeado de máquinas de última generación en HP, no estaba demasiado entusiasmado. Pero Woz estaba emocionado. Allí había treinta personas como él, personas obsesionadas quijotescamente con construir sus propios ordenadores. Cuando Marty Spergel repartió hojas de datos sobre el chip 8008, Woz se llevó una a casa y la examinó hasta que se dio cuenta de que esos miniordenadores que estaba pensando en diseñar (grandes máquinas como las que fabricaba Digital Equipment) eran innecesarios. Se podía hacer con microchips, como ese Altair que había visto esa noche. Consiguió toda la literatura que pudo sobre microprocesadores y escribió para obtener más información, abrió archivos sobre todo tipo de dispositivos de E/S y chips, y comenzó a diseñar circuitos para este futuro ordenador. El segundo boletín del Homebrew Computer Club publicó su informe sobre las actividades actuales:

> Que TVT haga mi propio diseño... Tengo mi propia versión de Pong, un videojuego llamado Breakthrough, un lector NRZI para casetes, ¡muy sencillo! Trabajo en una pantalla de ajedrez de TV de 17 chips (incluye 3 tableros almacenados); una pantalla de TV de 30 chips. Habilidades: diseño digital, interconexión, dispositivos de E/S, poco tiempo, tengo esquemas.

La atmósfera del Homebrew era perfecta para Steve Wozniak; había actividad y energía enfocada en la experimentación y la creatividad electrónica que eran tan esenciales para él como el aire que respiraba o la comida basura que comía. E incluso una persona que normalmente no era dada a socializar podía encontrarse haciendo amigos. Woz usaba a menudo su terminal de casa para acceder a la cuenta que se había creado para los miembros de Homebrew en el servicio Call Computer. (Call Computer era un servicio que permitía a las personas con terminales de casa acceder a una computadora central por teléfono). Había un programa en la computadora muy parecido a la función en el sistema ITS del MIT, donde dos personas podían "chatear" entre sí mientras estaban en la computadora, compartiendo información. Woz no sólo utilizó esto para comunicarse electrónicamente con la gente, sino que también hackeó las profundidades del sistema y descubrió una forma de entrar en las conversaciones electrónicas de otras personas. Así, cuando Gordon French, por ejemplo, estaba entusiasmado con su nuevo truco con el 8008 Chicken Hawk, su terminal de casa inexplicablemente comenzaba a imprimir estos chistes polacos semi-obscenos, y nunca se dio cuenta de que en algún lugar a kilómetros de distancia Steve Wozniak estaba doblado de risa.

Woz también conoció a Randy Wigginton, un chico de catorce años, rubio y atlético, que había logrado conseguir un trabajo en Call Computer. Wigginton vivía justo al final de la calle del desordenado apartamento con jardín que Wozniak compartía con su esposa, y Woz llevaba al joven a las reuniones de Homebrew. Desde antes de la escuela secundaria, Wigginton había estado enamorado de las computadoras. Llegó a idolatrar a Woz por su profundo conocimiento de las computadoras y apreciaba profundamente el hecho de que Woz, de veinticinco años, "hablara con cualquiera sobre cualquier tema técnico", incluso con un chico de catorce años como Wigginton. Aunque los padres de Randy estaban preocupados por el hecho de que las computadoras estaban tomando el control de la vida de su hijo, su obsesión se profundizó, alimentada por la enseñanza informal de Woz en el restaurante Denny’s de Foothill Drive, cuando volvían de una reunión, se reunían en el destartalado Malibu de Woz, con sus montañas de basura en el asiento trasero (docenas de bolsas de McDonald’s y revistas técnicas, todas empapadas por la extraña renuencia de

Woz a subir las ventanillas cuando llovía) y paraban a comprar Coca-Cola, patatas fritas y aros de cebolla. “Le hacía a Woz cualquier pregunta tonta para que hablara (‘¿Cómo funciona un intérprete BASIC?’) y lo escuchaba mientras hablaba”, recordó Wigginton más tarde.

Wozniak pronto conoció a otro miembro de Homebrew que trabajaba en Call Computer: John Draper. John Draper, un ingeniero a tiempo parcial, era más conocido como el “Capitán Crunch”, el héroe “phreak telefónico” de aquel artículo de Esquire que entusiasmó a Woz en 1971. Draper, cuya voz no modulada podía zumbarse como los últimos gemidos de una alarma de incendios, un hombre desaliñado que nunca parecía peinarse su largo cabello oscuro, recibió ese apodo después de descubrir que cuando uno hacía sonar el silbato que venía en el cereal de desayuno con ese nombre, el resultado sería el tono preciso de 2.600 ciclos que la compañía telefónica usaba para enviar el tráfico de larga distancia por las líneas telefónicas. John Draper, entonces aviador destinado en el extranjero, utilizó este conocimiento para llamar a sus amigos en casa.

Pero el interés de Draper iba más allá de las llamadas gratuitas: como ingeniero con una tendencia latente de hacker hacia la exploración que pronto resultaría abrumadora, se fascinó con el sistema de la compañía telefónica. “Hago [phreaking] por una razón y sólo por una razón”, le dijo al periodista de Esquire que lo hizo famoso en 1971. “Estoy aprendiendo sobre un sistema. La compañía telefónica es un sistema. Una computadora es un sistema. ¿Entiendes? Si hago lo que hago, es sólo para explorar un sistema. Eso es lo mío. La compañía telefónica no es más que una computadora”. Era la misma fascinación que compartían los hackers del Tech Model Railroad Club, en particular Stew Nelson (el hacker del MIT que había hackeado teléfonos desde la infancia); pero, al no tener el acceso de Nelson a herramientas sofisticadas para explorarlo, Draper tuvo que idear sus propios medios improvisados de acceso. (La única vez que Nelson conoció a Draper, el hacker del MIT no se impresionó por la habilidad técnica de Draper). Draper recibió ayuda al descubrir una red de phreaks telefónicos con intereses similares, muchos de ellos hombres ciegos que podían identificar fácilmente los tonos que podían hacer pasar a uno por el sistema. Draper se quedó atónito al descubrir que existían sistemas telefónicos alternativos desde los que se podía acceder a los tableros de pruebas, a los troncos de verificación para entrar en las conversaciones de la gente (una vez asustó a una mujer que le gustaba interrumpiendo furiosamente su conversación telefónica con otro hombre) y a las unidades de conmutación en el extranjero. Pronto descubrió cómo saltar de un circuito a otro y dominó los secretos de las “cajas azules”, que, como el ajuste de Stew Nelson al PDP-1 una década antes, podían enviar tonos a través de las líneas telefónicas para obtener llamadas de larga distancia ilimitadas y gratuitas.

Pero John Draper, que a veces actuaba tan impulsivamente que parecía un niño grande que berrincheaba por la leche de conocimientos de sistemas de su madre, no tenía la determinación centrada de los hackers del MIT: se le podía engatusar fácilmente para que entregara la información sobre las cajas azules a personas que anhelaban vender las cajas a personas que querían llamadas gratis, como habían hecho Wozniak y Jobs de puerta en puerta en los dormitorios de Berkeley.

Las propias incursiones telefónicas de Draper fueron más benignas. Un típico delito era buscar y “mapear” varios códigos de acceso para países extranjeros, y utilizar esos códigos para saltar de una línea troncal a otra, escuchando una serie de clics edificantes mientras su señal rebotaba de un satélite de comunicaciones al siguiente. Sin embargo, después del artículo de Esquire, las autoridades lo persiguieron, y en 1972 fue atrapado en el acto de llamar ilegalmente a un número de Sydney, Australia, que proporcionaba a los que llamaban los nombres de las mejores canciones de Down Under. Por esta primera infracción, fue condenado a una pena suspendida.

Se dedicó a la programación informática y pronto se convirtió en un hacker habitual. La gente lo recordaría más tarde en las comidas compartidas de People's Computer Company, llenando su plato hasta el cielo y atiborrándose. Era un antitabaco virulento y también gritaba casi dolorosamente cuando alguien encendía un cigarrillo. Todavía estaba interesado en el phreaking, y entre los temas de los que hablaba en las comidas compartidas estaban cosas como obtener acceso a ARPAnet, algo que consideraba eminentemente justificable: "Tenía algunas integraciones que tenía que hacer analíticamente. El ordenador del MIT [tenía un programa para ayudarme a hacerlo]. Así que lo usé", explicaría más tarde.

Cuando terminaron las comidas compartidas, gravitó hacia Homebrew. Era consultor de Call Computer y había hecho los arreglos para que el Homebrew Club obtuviera su cuenta. Se convirtió en un gran fanático del hacking de Wozniak, y Wozniak estaba emocionado de conocer al famoso phreak telefónico que había inspirado sus propias escapadas de la caja azul. No era inusual verlos juntos en el fondo de la sala, como lo estaban una noche a fines de 1975 cuando Dan Sokol se acercó a ellos. Sokol era el chico rubio de pelo largo que se ponía de pie en el Homebrew, verificaba que nadie de Intel estuviera cerca, y canjeaba chips 8080 con cualquiera que tuviera un buen equipo para intercambiar.

En ese momento, Sokol se estaba arruinando por usar su terminal de casa para acceder a la cuenta de Call Computer. Como Sokol vivía en Santa Cruz y Call Computer se encontraba en Palo Alto, su factura telefónica era escandalosa; accedía a la computadora entre cuarenta y cincuenta horas a la semana. La solución llegó un día en la parte trasera del auditorio de SLAC cuando Sokol fue presentado a Wozniak y John Draper.

¿No eres el Capitán Crunch?

"¡Sí, ese soy yo!", se ofreció Draper, y Sokol inmediatamente lo acribilló a preguntas sobre cómo construir una caja azul, que le permitiría hacer las llamadas telefónicas de Santa Cruz a Palo Alto gratis. Aunque la libertad condicional de Draper especificaba que se negaba a divulgar sus secretos de hacking telefónica, no pudo resistirse cuando la gente le preguntó; el hacker en su sangre simplemente dejó que la información fluyera. "En los quince minutos siguientes, procedió a decirme todo lo que necesitaba saber [para construir una caja azul]", dijo Sokol más tarde. Pero cuando Sokol armó la caja azul, no funcionó; se lo hizo saber a Draper y el sábado siguiente, Draper, acompañado por Steve Wozniak, vendría. Revisaron la caja de Sokol. "Parece estar bien", dijo Draper, y comenzaron a ajustar los tonos de oído. Esta vez, cuando Sokol probó la caja azul, funcionó. Sokol usaría la caja sólo para conectarse a la computadora, una práctica que en la mente de los hackers justifica la infracción de la ley, y no para obtener ganancias personales en asuntos triviales como llamar a parientes lejanos.

Wozniak echó un vistazo al "kluge" de Sokol, la computadora que había obtenido mediante el trueque de piezas liberadas, y ambos lamentaron el alto costo de hackear el hardware. Woz se quejó de que, aunque trabajaba para Hewlett-Packard, los vendedores no querían desprenderse de ningún chip para él. En la siguiente reunión de Homebrew, Dan Sokol le presentó a Wozniak una caja llena de piezas que funcionarían con un microprocesador Motorola 6800. Woz recibió un manual del 6800 y comenzó a diseñar un ordenador que pudiera interactuar con la TV Typewriter que había construido. Cuando alguien llevó a una reunión de Homebrew un ordenador que tenía vídeo incluido, él sabía que su ordenador también tendría que tener vídeo incorporado. Le gustaba la idea de un ordenador en el que se pudiera jugar a un videojuego. En esa época se estaba celebrando la

feria de ordenadores de Wescon y Wozniak pasó por el stand de MOS Technology y descubrió que estaban vendiendo los primeros modelos de su nuevo chip microprocesador, el 6502, por sólo veinte dólares. Como el chip no era muy diferente del Motorola 6800, compró un puñado y decidió que el 6502 sería el corazón de su nueva máquina.

Wozniak no estaba pensando en construir un ordenador para vender. Estaba construyendo un ordenador para divertirse, para mostrarle a sus amigos. Le comentaba lo que estaba haciendo a su amigo Steve Jobs de Atari, que estaba interesado en los terminales y estaba pensando en crear una empresa que los fabricara. Cada dos semanas, Woz iba a Homebrew para ver o escuchar las novedades, y nunca tenía ningún problema en seguir los detalles técnicos porque todo el mundo tenía libertad para dar información. Algunas cosas las incorporaba al ordenador; por ejemplo, cuando vio la placa Dazzler, supo que quería gráficos en color. Sabía, por supuesto, que quería un BASIC, y como el único BASIC que funcionaba en el 6502 en aquel momento era el Tiny BASIC de Tom Pittman, y Woz quería un BASIC “grande”, escribió el suyo propio. Le daba el código a todo el que lo quería e incluso publicaba algunas de sus subrutinas en el Diario del Dr. Dobbs.

Cuando terminó, tenía un ordenador que en realidad no era un kit ni un ordenador ensamblado, sino una placa cargada con chips y circuitos. Con esa placa no se podía hacer nada, pero si se conectaba a la placa una fuente de alimentación, un teclado, un monitor de vídeo y un reproductor de casetes, se conseguía un ordenador funcional con pantalla de vídeo, almacenamiento masivo y entrada/salida. A continuación, se podía cargar el «Integer BASIC» de Steve Wozniak y escribir programas. Su ordenador tenía varias cosas sorprendentes, y una de ellas, y no la menor, era que había reunido la potencia y las capacidades de un Altair y varias placas en una placa mucho más pequeña. Lo que a otros les hacía falta con dos chips, Woz lo hacía con uno. Esto no sólo era prudente desde el punto de vista fiscal, sino una especie de machismo técnico que recordaba a la obsesión por el código en los días de TMRC, cuando Samson, Saunders y Kotok intentaban reducir una subrutina a la menor cantidad de instrucciones.

Más tarde, Wozniak explicó por qué la placa utilizaba tan pocos chips: «Me gusta por motivos estéticos y me gusta considerarme inteligente. Ése es mi problema, y hago diseños que utilizan un chip menos que el anterior. Pensaba cómo podría hacerlo más rápido, más pequeño o de forma más inteligente. Si [trabajo en algo] que se considera un buen trabajo utilizando seis instrucciones, lo intento en cinco o tres, o dos si quiero ganar [a lo grande]. Hago cosas complicadas que no son normales. Cada problema tiene una mejor solución cuando empiezas a pensar en él de forma diferente a la habitual. Y los veo, todos los días veo varios problemas, me pregunto si es un problema de hardware. Empiezo a analizar muchas técnicas que he utilizado antes, contadores y retroalimentación o registros de chips... un enfoque de línea de base, buscando pequeños puntos finales específicos de una jerarquía... crea básicamente una especie de matemática diferente. Los descubrimientos aumentaron mi motivación porque tendría algo que mostrar y esperaba que otras personas los vieran y dijeran: "Gracias a Dios, así es como quiero hacerlo", y eso es lo que obtuve del Homebrew Club".

Wozniak trajo la placa, junto con el hardware para hacerla funcionar al Homebrew. No tenía una grabadora de casetes, y mientras la reunión continuaba, se sentó afuera, escribiendo frenéticamente el código hexadecimal (3000 bytes) del intérprete BASIC de 3K en la máquina. Ejecutaba una prueba en parte del programa, y la prueba podría fallar y él comenzaría de nuevo. Finalmente, estaba funcionando, aunque era sólo una versión preliminar que no tenía el conjunto completo de

comandos, y cuando la gente pasaba por encima, Wozniak explicaba, con su zumbido de alta velocidad y sin aliento, lo que podía hacer.

No pasó mucho tiempo antes de que Wozniak se dirigiera a todo el Homebrew Computer Club, sosteniendo su plaqueta en el aire y respondiendo a las preguntas de los miembros, la mayoría de los cuales le preguntaban cómo hacía esto o si iba a incorporar esta o aquella característica. Eran buenas ideas, y Wozniak llevaba su equipo cada dos semanas, se sentaba en la parte trasera del auditorio donde estaba la toma de corriente, recibía sugerencias para mejoras e incorporaba esas mejoras.

El amigo de Woz, Steve Jobs, estaba muy entusiasmado con la placa; pensaba que, como Processor Technology y Cromemco, deberían fabricar las placas en cantidad y venderlas. Jobs, a los veintidós años, era un par de años más joven que Wozniak, y no mucho más limpio. Jobs tenía lo que se describía como una "barba a lo Fidel Castro", a menudo iba descalzo y tenía un interés californiano por las filosofías orientales y el vegetarianismo. Era un promotor incansable, de lengua de oro y hábil persuasor. Pronto a los dos se les conoció como "los dos Steves", y el ordenador de Wozniak era conocido como Apple, un nombre concebido por Jobs, que en su día trabajó en un huerto. Aunque la dirección oficial de la empresa Apple - que todavía no se había constituido como sociedad anónima - era un buzón de correo: Jobs y Wozniak trabajaban en realidad en un garaje. Para conseguir capital, Jobs vendió su combi Volkswagen y Woz vendió su calculadora programable HP. Jobs puso anuncios en publicaciones para aficionados y empezaron a vender Apples al precio de 666,66 dólares. Cualquiera que estuviera en el Homebrew podía echar un vistazo a los esquemas del diseño, el BASIC de Woz se regalaba con la compra de un equipo que conectaba el ordenador a una grabadora de casetes, y Woz publicó las rutinas para su "monitor" 6502, que permitía analizar la memoria y ver qué instrucciones estaban almacenadas, en revistas como Dr. Dobbs. El anuncio de Apple incluso decía: "nuestra filosofía es proporcionar software para nuestras máquinas de forma gratuita o a un coste mínimo".

Mientras se realizaba la venta, Steve Wozniak empezó a trabajar en un diseño ampliado de la placa, algo que impresionaría aún más a sus colegas del Homebrew. Steve Jobs tenía planes de vender muchos ordenadores basados en este nuevo diseño, y empezó a conseguir financiación, apoyo y ayuda profesional para el día en que el producto estuviera listo. La nueva versión del ordenador de Steve Wozniak se llamaría Apple II, y en aquel momento nadie sospechaba que se convertiría en el ordenador más importante de la historia.

• • • • • • •

Fue la atmósfera fértil del Homebrew la que guió a Steve Wozniak a través de la incubación del Apple II. El intercambio de información, el acceso a consejos técnicos esotéricos, la energía creativa arremolinada y la oportunidad de dejar a todos boquiabiertos con un diseño o programa bien hackeado... estos fueron los incentivos que sólo aumentaron el intenso deseo que Steve Wozniak ya tenía: construir el tipo de ordenador con el que quería jugar. La informática era el límite de sus deseos; no lo acosaban visiones de riqueza y fama, ni estaba obsesionado por sueños de un mundo de usuarios finales expuestos a ordenadores. Le gustaba su trabajo en HP y le encantaba la atmósfera embriagadora de estar rodeado de los ingenieros caballerosos que estaban en la cima de la industria informática. En un momento dado, Wozniak preguntó a sus jefes de HP si querían que diseñara el ordenador Apple para ellos; pensaron que no era comercializable y le dieron autorización para venderlo por su cuenta. Cuando pareció que HP iba a crear una pequeña división

informática, Wozniak solicitó un traslado; pero, según Alan Baum, "el jefe del laboratorio no estaba impresionado. No tenía título universitario" (Woz había dejado Berkeley antes de graduarse).

Así que trabajó en el Apple II, a menudo hasta las cuatro de la mañana; pronto sería un miembro más del Homebrew divorciado por una viuda de un informático. Diseñar el Apple II no fue nada fácil. Hubo cientos de problemas para hacer una combinación de computadora y terminal autónoma y lista para programar; Wozniak ni siquiera tenía los moderados recursos y el flujo de caja que tenían Bob Marsh y Lee Felsenstein cuando diseñaron la Sol, la primer combinación de computadora y terminal de entrada y una de las muchas inspiraciones para el Apple II. Pero tenía una visión de lo que quería que fuera su computadora y podía recurrir a la ayuda de Homebrew y otros expertos en Silicon Valley. Finalmente, tenía un prototipo funcionando. Él y Randy Wigginton lo llevaron (un revoltijo suelto pero completamente conectado de piezas y placas) a una reunión de Homebrew en diciembre de 1976 en un par de cajas, junto con un televisor en color Sears destartalado.

Años después, las personas que asistieron a esa reunión de Homebrew recordarían diferentes versiones de la reacción a la presentación de Stephen Wozniak del Apple II. Wozniak y los demás fanáticos del chip 6502 salieron con la impresión de que la computadora había emocionado a todos. Otros pensaron que era simplemente un avance más en la frenética escalada hacia una computadora casera definitiva. Como dijo Lee Felsenstein, "La gente del Homebrew no se quedó sentada esperando que Apple se hiciera realidad: la gente estaba haciendo cosas, hablando de cosas, mostrando cosas".

Una cosa que no entusiasmaba a los miembros era el hecho de que los modelos de producción del Apple sólo se venderían en forma totalmente ensamblada: ¿para qué comprar una computadora, pensaban los hackers de hardware, si no lo podían construir ellos mismos? Los más acérrimos de la vieja escuela, que respetaban la solidez y previsibilidad de los productos de Processor Technology y Cromemco, pensaban que la Apple era interesante, especialmente por su circuito económico y sus capacidades de color, pero no era una máquina tan buena como la Sol, que se basaba en el conocido bus Altair (recientemente bautizaado bus S-100 por consenso de fabricantes, en particular Marsh y Garland, que estaban hartos de referirse a una parte de sus ordenadores con el nombre de un competidor que, en un espíritu poco hacker, parecía resentir su existencia). La Apple tenía un bus completamente nuevo y un sistema operativo totalmente nuevo, ambos diseñados por Woz; además, tenía el desconocido chip 6502 como cerebro. Además, una empresa probada como Processor Technology parecía más propensa a ser capaz de dar soporte a una máquina en el campo que Apple, que aparentemente consistía sólo en dos barbudos en un garaje.

En el fondo, sin embargo, el desacuerdo se reducía a cuestiones religiosas de diseño. La Sol reflejaba los temores apocalípticos de Lee Felsenstein, moldeados por la ciencia ficción post-holocausto, de que la infraestructura industrial podría desaparecer en cualquier momento y la gente debería poder conseguir piezas para mantener su máquina en funcionamiento entre los escombros de esta sociedad devastada; idealmente, el diseño de la máquina sería lo suficientemente claro como para permitir a los usuarios averiguar dónde colocar esas piezas.

“Tengo que diseñar de manera que se pueda armar a partir de tachos de basura”, dijo una vez Felsenstein. “En parte porque así es como empecé, pero sobre todo porque no confío en la estructura industrial: podrían decidir reprimirnos a nosotros, los raros, y tratar de negarnos las piezas que necesitamos”. Esta filosofía se expresó en el VDM y en el propio Sol, ambos productos

que hacían su trabajo limpiamente, de una manera no demasiado llamativa y con una falta proletaria de sentimentalismo.

La Apple de Steve Wozniak fue otra historia. Wozniak, que creció en una familia convencional en el protegido mundo suburbano de California, con casas unifamiliares, ferias científicas y hamburguesas de McDonald's, tenía una seguridad innata. Se sentía cómodo asumiendo riesgos, dejando que el diseño fuera tan lejos como su imaginación lo permitiera. Creó una maravilla estética optimizando un número limitado de componentes electrónicos disponibles comercialmente de modo que, dispuestos y cableados de manera muy ingeniosa, no solo ofrecieran la potencia de un PDP-1, sino también color, movimiento y sonido.

Si Woz pudiera hacer las cosas a su manera, agregaría funciones para siempre. Tan solo dos días antes de la reunión, había modificado la máquina para que pudiera mostrar gráficos especiales en color de alta resolución. Lo hizo no mediante la forma habitual de agregar chips especiales para hacerlo, sino ideando una forma de conectar la máquina para que la unidad central de procesamiento, la 6502, pudiera cumplir una doble función.

El genio de Woz para la optimización a veces tenía efectos extraños. Por ejemplo, la forma en que Apple llenaba la pantalla con una imagen era muy diferente del método de Sol, que llenaba las cosas en un orden adecuado; Apple dibujaba su pantalla de una manera aparentemente aleatoria, como un mosaico desorganizado. No lo hacía por casualidad, sino porque Woz se dio cuenta de que al hacerlo de esa manera se ahorraría una instrucción por cada línea que apareciera en la pantalla. Un truco inteligente, desdeñado por algunos que pensaban que era indicativo de la imprevisibilidad y la "falta de precisión" de Apple, pero muy admirado por aquellos que podían apreciar la belleza de un diseño maximizado. En conjunto, el diseño reflejaba un tour de force de hacking, y un ingeniero muy astuto podía ver los ingeniosos giros de la trama, los optimistas vuelos de la imaginación y los chistes excéntricamente cósmicos encarnados en la máquina.

Una persona que pensaba que el Apple II era simplemente genial era Chris Espinosa, un joven conocido de Randy Wigginton. Espinosa era un chico de secundaria de catorce años, delgado y pálido, que amaba las computadoras y reprobaba las clases de matemáticas porque sentía que hacer la tarea no era un uso óptimo del tiempo. Quedó fascinado con el ordenador de Steve Wozniak. A partir de la explicación de la sintaxis de los comandos BASIC especiales de Woz que surgieron en la charla y de la explicación de los bocetos de las entrañas de la máquina distribuidos por todas partes, Espinosa anotó algunos programas BASIC y, durante el período de acceso aleatorio de la reunión, cuando la gente se agolpaba alrededor de la nueva máquina, tomó el control del teclado y tecleó frenéticamente algunos programas que crearon llamativas pantallas en color en el viejo y gran televisor Sears que Wozniak había arrastrado. Wozniak estaba emocionado: "No pensé que alguien más pudiera venir y mostrarme: '¡Mira!', y entusiasmarse y mostrarle a otras personas y decir: 'Mira, esto es fácil, simplemente introduces este comando y haces esto'". Allí estaba este chico de secundaria, ejecutando programas en este pequeño ordenador que Wozniak había construido. La reacción de Steve Jobs fue más pragmática: contrató a Chris Espinosa como uno de los primeros empleados de la empresa. Al igual que el otro especialista en software adolescente, Randy Wigginton, ganaría tres dólares por hora.

Steve Jobs se concentraba a tiempo completo en desarrollar la compañía Apple para estar lista para lanzar el Apple II al año siguiente y causar un gran revuelo en el mercado. Jobs era un orador brillante que, según Alan Baum, "se esforzaba al máximo... me contaba los precios que conseguía

por las piezas, y eran favorables a los precios que pagaba HP". Como ingeniero, Jobs era mediocre; su punto fuerte era el de planificador, alguien con visión para ver cómo las computadoras podían extenderse hasta un punto de utilidad más allá del soñado por hackers puros como Steve Wozniak. También era lo suficientemente inteligente para darse cuenta de que, como un joven de veintidós años de pelo largo cuya vestimenta habitual eran vaqueros y pies descalzos, no era la persona adecuada para dirigir una gran corporación informática; sobre todo, carecía de experiencia en gestión y marketing. Decidió que contrataría talentos de gestión de primera categoría y muy caros para dirigir Apple Computer.

No era una conclusión fácil en aquellos días, cuando ingenieros como Ed Roberts y Bob Marsh pensaban que construir una máquina de calidad era el ingrediente principal para el éxito, y que la gerencia podría cuidar de sí misma. Ed Roberts aprendió lo absurdo de esto, a las malas. A mediados de 1976, Roberts estaba cansado de la "telenovela" (en sus palabras) en la que se había convertido MITS, con clientes frustrados, una línea confusa de varias versiones nuevas y mejoradas del Altair, cientos de empleados, una política interna viciosa, distribuidores perpetuamente en pánico, finanzas desesperadamente confusas y ni una sola noche de sueño decente en más de un año. Había estado diseñando un nuevo y emocionante ordenador Altair 2 (una máquina compacta y de alta potencia que podía caber en un maletín), pero la mayor parte de sus energías se gastaban en disputas de gestión. Así que decidió llamarlo, dijo más tarde, "una página en mi vida; era hora de pasar a la siguiente página", y sorprendió al mundo de los hackers de hardware al vender la empresa a una gran firma llamada Pertec. A finales de año, Roberts, con su compra de más de un millón de dólares, dejó el negocio y se convirtió en granjero en el sur de Georgia.

La moraleja de la historia era que los ingenieros no necesariamente pueden dirigir empresas. Pero encontrar gente que pueda hacerlo no es fácil, especialmente cuando tu empresa, al menos en la superficie, parece un pequeño aquelarre de hippies y estudiantes de secundaria. Chris Espinosa señaló más tarde que, a principios de 1977, Jobs parecía tan desaliñado que "no lo dejaban subir a minibuses y aviones, mucho menos entrar en los pasillos del poder de la industria de los semiconductores", pero logró un gran golpe al incorporar a Mike Markkula al equipo de Apple. Markkula era un ex genio del marketing, ahora de unos treinta y cinco años, que se había retirado de Intel unos años antes; Desde entonces, Jobs se dedicó a diversas actividades, algunas relacionadas con los negocios, otras tan extrañas como inventar un diagrama de rueda para mostrar las distintas posiciones de los dedos en los acordes de guitarra. Jobs le pidió que le ayudara a diseñar un plan de negocios para Apple, y Markkula terminó ayudando a conseguir capital de riesgo para la empresa y firmando como su primer presidente del consejo de administración. Fue a través de Markkula que Jobs también consiguió que un gerente de Fairchild Semiconductor llamado Mike Scott se convirtiera en presidente de la empresa. De modo que, mientras la empresa más importante con un terminal-ordenador en el mercado, Processor Technology, luchaba con la gestión inexperta de los hackers de hardware Bob Marsh y Gary Ingram, Apple estaba preparada para crecer.

Esta actividad del mundo real realmente no había calado en lo que respecta a Steve Wozniak. Chris Espinosa y Randy Wigginton venían a su casa después de jugar con la versión a medio construir del Apple II de Wigginton, y allí, en el piso de la sala de estar del pequeño lugar de Woz, depuraban programas y hardware, escribían programas de generación de tonos, soldaban placas. Era divertido. Mientras tanto, en su propio garaje, Jobs dirigía las operaciones diarias. "Venía de vez en cuando y veía lo que estábamos haciendo, hacía recomendaciones, pero no hacía ningún diseño", dijo Espinosa más tarde. "Emitía juicios - que es su mayor talento - los teclados, el diseño de la carcasa,

el logotipo, qué piezas comprar, cómo diseñar la placa de circuito impreso para que se viera bien, la disposición de las piezas, los distribuidores que elegíamos... el método de montaje, el método de distribución, todo".

En esto lo guiaba la mano experimentada de Mike Markkula, que se tomaba muy en serio la aventura de Apple. Una cosa que aparentemente reconoció fue que el compromiso de Steve Wozniak era con el ordenador más que con la empresa. Para Woz, la Apple era un hack brillante, no una inversión. Era su arte, no su negocio. Recibía su pago resolviendo acertijos, ahorrando chips, impresionando a la gente de Homebrew. Esto estaba bien para el hacking, pero Markkula quería, al menos, la participación a tiempo completo de Wozniak en la empresa. Le dijo a Jobs que le dijera a su socio que si Woz quería que hubiera una empresa Apple Computer, debía dejar HP para trabajar a fondo en la preproducción del Apple II.

Fue una decisión difícil para Wozniak. "Esto fue diferente del año que pasamos montando el Apple I en el garaje", recordó más tarde Wozniak. "Esta era una empresa real. Diseñé un ordenador porque me gusta diseñar, para presumir en el club. Mi motivación no era tener una empresa y ganar dinero. Mike me dio tres días para decir si iba a dejar HP, sí o no. Me gustaba HP. Era una buena empresa y yo estaba seguro y había mucho trabajo bueno. No quería irme y dije que no".

Steve Jobs escuchó la decisión y llamó a los amigos y familiares de Wozniak, rogándoles que convencieran a Woz de que dejara HP y trabajara para Apple a tiempo completo. Algunos de ellos lo hicieron, y cuando Woz escuchó los argumentos, reconsideró. ¿Por qué no trabajar para que el Apple II saliera al mundo? Pero incluso cuando aceptó dejar HP y trabajar con Jobs a tiempo completo, se dijo a sí mismo que lo que estaba haciendo ya no era hacking puro. La verdad era que crear una empresa no tenía nada que ver con el hacking o el diseño creativo. Se trataba de ganar dinero. Fue "cruzar la línea", como dijo más tarde Wozniak. No se trataba de una estafa: Wozniak creía en su computadora y tenía confianza en el equipo que la produciría y vendería, pero "no hay forma de que yo asociara a Apple con hacer un buen diseño de ordenadores en mi cabeza. No fue esa la razón por la que puse en marcha Apple. La razón por la que puse en marcha Apple después del diseño del ordenador es diferente: ganar dinero".

Fue una decisión crucial que simbolizaría el cambio que se estaba produciendo en los ordenadores pequeños. Ahora que hackers como Wozniak construían máquinas con terminales y teclados, máquinas que presumiblemente podrían ser útiles para gente que no fueran aficionados, la dirección de la industria en ciernes ya no estaba en manos de esos hackers. Habían pasado casi veinte años desde que los hackers del TMRC conocieran el TX-0. Ahora, entrar en el negocio era lo correcto.

En enero de 1977, la media docena de empleados de esta nueva empresa - que no se constituiría hasta marzo de ese año - se trasladaron a un espacio reducido en Stevens Creek Boulevard en Cupertino, a tiro de piedra de un 7-Eleven y un restaurante de comida sana Good Earth. Wozniak prefería caminar por la calle para ir a Bob's Big Boy. A primera hora de la mañana, él y Wigginton iban allí, pedían una taza de café, tomaban un sorbo y hablaban de lo malo que estaba el café, dejando la taza casi llena sobre la mesa. Era una especie de ritual. A Woz le gustaba coger paquetes de Fizine, un antiácido burbujeante, y vaciarlos en los azucareros de Bob's, donde esperaba hasta que algún cliente desprevenido ponía lo que él creía que era azúcar en su café. Estallaba como un pequeño volcán y Woz se divertía mucho. Pero a menudo Woz se limitaba a hablar, sobre todo de cosas técnicas, y a veces sobre Apple. Wigginton y Espinosa, ambos todavía en la escuela secundaria, se habían tomado en serio algunas de las hipérboles de Jobs, al estilo de los

planificadores, todos lo habían hecho en cierta medida, y creían que la cruzada Homebrew se centraba precisamente allí, en Stevens Creek Boulevard. “Todo el mundo estaba muy metido en ello”, dijo Wigginton más tarde. “Estábamos más motivados por un sueño de lo que iba a suceder que por lo que realmente estaba sucediendo. Que seríamos una empresa exitosa y que íbamos a sacar al mercado el mejor producto que se había producido nunca”.

A menudo trabajaban las veinticuatro horas del día, soldando, diseñando y programando. Uno de los amigos de Woz contratado como especialista en hardware hacía reclamos de pájaros mientras trabajaba. Woz hacía bromas, jugaba y luego hacía una increíble cantidad de trabajo en un breve período. Woz y sus amigos estaban preparando un tipo de computadora diferente a los superventas anteriores, la Altair, la Sol y la IMSAI. Steve Jobs y Mike Markkula sentían que el mercado de Apple iba mucho más allá de los aficionados y que la máquina tenía un aspecto más amigable. Jobs contrató a un diseñador industrial para que construyera una carcasa de plástico elegante y de perfil bajo en un cálido color beige terroso. Se aseguró de que el diseño de Woz fuera atractivo una vez que se levantara la tapa de la carcasa. El bus de Apple, al igual que el bus S-100, era capaz de aceptar placas de circuito adicionales para que hiciera cosas interesantes, pero Woz había seguido el consejo de su amigo Alan Baum y había hecho que las ocho "ranuras de expansión" dentro de la Apple fueran especialmente fáciles para que los fabricantes pudieran hacer placas de circuito compatibles. Por supuesto, les ayudaría la arquitectura "abierta" de la máquina; fiel a la ética hacker, Woz se aseguró de que la Apple no tuviera secretos que impidieran a la gente crear en ella. Cada giro y vuelta de su diseño, cada truco de codificación en su intérprete BASIC (que se incluiría dentro de esta máquina, cableado en un chip de circuito personalizado) sería documentado y distribuido a cualquiera que quisiera verlo.

En ciertos puntos, Woz y Jobs confiaron en sus conexiones del Homebrew para obtener ayuda. Un buen ejemplo fue lo que sucedió con un problema potencial para obtener la aprobación de la FCC para la computadora. Rod Holt - un ingeniero de Atari que había estado ayudando a diseñar la fuente de alimentación - declaró con tristeza que el conector de la máquina a un televisor (llamado modulador de radiofrecuencia [RF]) emitía demasiadas interferencias y nunca pasaría la prueba de la FCC. Así que Steve Jobs recurrió a Marty Spergel, el hombre de la chatarra.

Spergel solía aparecer en las reuniones del Homebrew, sosteniendo alguna pieza esotérica y regalándola. “Miraba mi caja de chatarra y decía: ‘¡Aquí tengo una caja llena de la A, B, C!’, y la gente corría a mil kilómetros por hora y antes de que pudiera soltar la caja, estaba vacía”. Tenía olfato para los nichos del mercado de la electrónica y recientemente había hecho una fortuna importando controladores de joystick de Hong Kong para que la gente pudiera jugar a juegos como el Target de Steve Dompier en Altairs y Sols. En un momento dado, su empresa - M&R Electronics - incluso presentó un kit de computadora, pero ese producto nunca llegó a tener éxito. Un día, Marty visitó la sede de Apple en Cupertino, que constaba de una sola sala, y habló con Woz, Jobs y Rod Holt sobre la situación de los moduladores. Estaba claro que Apple no podía comercializar los ordenadores con los moduladores actuales, por lo que se decidió que Holt le daría a Marty Spergel las especificaciones de los moduladores y que él los fabricaría. “Mi parte consistía en mantener a la FCC alejada de Apple Computer”, dijo Spergel más tarde. “Así que lo que hice fue enviar los moduladores desde mi casa, y Apple enviaba los Apple desde su fábrica. Pero cuando llegaban a los distribuidores, estos le vendían un modulador al usuario final, y cuando el usuario final [llegaba] a casa podía conectarle el modulador. En consecuencia, ahora es responsabilidad del usuario final [evitar las interferencias de RF]”.

Se trataba de un caso clásico de compartir en el Homebrew, en el que todos se beneficiaban para sortear un obstáculo burocrático. Spergel preguntó a Jobs cuántos moduladores - que M&R vendería bajo el nombre de “Sup’r Mod” por unos treinta dólares cada uno - serían necesarios. Jobs prometió que se venderían en grandes cantidades, quizás hasta cincuenta unidades al mes.

Varios años después, Spergel calculó que había vendido cuatrocientos mil Sup’r Mods.

• • • • • • • •

A principios de 1977, Jim Warren, miembro del Homebrew Computer Club y editor del Dr. Dobbs Journal, estaba tramando un plan bastante grande. Warren era el tipo de pelo corto, cara ancha y barba que coleccionaba “tecnochismosos” como pasatiempo y veía a Homebrew como una salida para esparcir todo tipo de rumores sobre empresas en el “Silicon Gulch”, como él lo llamaba. A menudo, sus rumores eran ciertos. Además de sus deberes editoriales y sus actividades como yenta de silicio, Warren estaba en un “modo de disertación” en Stanford, como él mismo lo describía. Pero la tasa de crecimiento cuántico de la computadora personal le interesaba más que un doctorado. Era un fanático y consideraba que el movimiento de la computadora casera era una especie de festival de amor humanista post-Universidad Libre, en el que te quitas la ropa y te ensucias.

Su asistencia a la feria de computadoras PC '76 en Atlantic City había reforzado esa creencia. Al principio no había querido ir, considerando ese desvencijado centro turístico como "la entrepierna de la nación", pero el promotor de la feria lo había llamado y le había contado sobre todas las personas interesantes que estarían allí, agregando lo genial que sería para el editor de Dr. Dobbs estar presente, y Jim Warren se sintió algo frustrado porque, como Bob Albrecht le pagaba solo $350 al mes por editar la revista, tuvo que mendigar el dinero para el viaje. Pensó que la gran feria debería ser allí mismo, en California. Una noche estaba hablando con Bob Reiling, un ingeniero de Philco que silenciosamente había asumido las funciones de Fred Moore como editor del boletín Homebrew. Warren preguntó por qué demonios estaba sucediendo todo eso en la costa equivocada cuando el centro indiscutible del mundo de la microcomputadora estaba justo aquí. Reiling estuvo de acuerdo y Warren decidió que debían hacerlo, montar un espectáculo que también, en espíritu hacker, fuera un intercambio de información, equipos, conocimientos técnicos y buenas vibraciones. Podría tener la atmósfera idílica de la “Feria del Renacimiento” anual en el condado de Marin, una auténtica “Feria de Computadoras”.

Estaba pensando en este espectáculo cuando llegó a Atlantic City, que a pesar de la horrible humedad y el el hecho de que las instalaciones estuvieran en ruinas era, como diría más tarde, “una experiencia totalmente excitante. [Conocías] a todas las personas con las que habías hablado por teléfono o de las que habías recibido una carta y que estaban haciendo cosas... [te emocionaba] muchísimo conocer a las personas que estaban haciendo las cosas”. Estas reuniones cara a cara eran una nueva y poderosa característica de interfaz que proporcionaba una información mucho más actualizada que la que se podía obtener en las publicaciones. “El Dr. Dobbs tenía un plazo de entrega de seis semanas y eso me estaba volviendo loco. Diablos, seis meses era media generación de máquinas. La oportunidad de hablar con la gente sobre lo que estaban haciendo esa semana era una mejora radical. Así que fue en ese tipo de entorno que anuncié que íbamos a hacer una Feria de Computadoras en la Costa Oeste”.

Warren, con Reiling como socio, se propuso organizar el evento. Pronto se sintió intimidado por el hecho de que el lugar ideal, el Civic Auditorium de San Francisco, cobraba una tarifa de alquiler

considerable: ¡miles de dólares al día! Después de escuchar esto, Warren y Reiling se dirigieron a la península y se detuvieron en Pete's Harbor, un café al aire libre junto al puerto deportivo de la bahía, un lugar favorito de Albrecht y la gente del PCC. Warren recuerda: "Recuerdo que dije: 'Vaya, nos estamos metiendo en problemas. ¿Podemos permitirnos esto?'. Saqué una servilleta de un servilletero grande y comencé a garabatear. Cuántas exposiciones esperar. Cuántos asistentes. Si atrajeron a tres mil quinientos en Atlantic City, deberíamos duplicar esa cantidad... tal vez atraer hasta siete mil. ¿Cuánto cobrar por expositores y asistentes? Multiplíquelo. Súmelo...". Jim Warren se quedó atónito al descubrir que no sólo podían permitírselo, sino que además podían sacarle beneficios. Y, desde luego, no había nada malo en ello.

Jim Warren se puso al teléfono y empezó a llamar a los presidentes de las mayores empresas del sector, a la mayoría de los cuales conocía personalmente del Homebrew o de su trabajo en revistas. "Llamé a Bob Marsh y le dije: 'Oye, vamos a hacer una feria de ordenadores, ¿te interesa?' y él dijo: 'Claro que sí: 'Vale, envíanos algo de dinero y te conseguiremos un espacio de exposición. Muy lejos'. Llamamos a Harry Garland de Cromemco. 'Soy Jim Warren, vamos a hacer una feria de ordenadores. ¿Quieres participar?' 'Claro, de acuerdo'. 'Sí, bueno, te enviaremos un plano de stands en cuanto tengamos la oportunidad. Envíanos el dinero porque lo necesitamos'. Creo que tardamos cuatro días en tener números negros".

Warren resultó tener un talento considerable como promotor. Comenzó un periódico sensacionalista específicamente para generar entusiasmo sobre la Feria y, de paso, para difundir su marca de chismes tecnológicos. Se llamó Silicon Gulch Gazette, y había historias sobre cómo sería la Feria y pequeños perfiles de algunos de los oradores, y también un perfil de Jim Warren, el "presidente de la silla". El periódico se jactaba de los acuerdos de "copatrocinio" de la Feria con grupos sin fines de lucro como el Homebrew Computer Club, SCCS, PCC y su filial, Community Computer Center (CCC), y otros. (Joanne Koltnow, que ayudó a la Feria desde su trabajo en CCC, dijo más tarde que "todo el mundo se sorprendió" cuando descubrieron más tarde que la Feria era una organización con fines de lucro). Con un personal de dos secretarias, Warren y sus socios trabajaron casi las veinticuatro horas del día a medida que avanzaba la Feria.

También trabajaron frenéticamente antes de la Feria los ocho empleados de Apple Computer. Apple había alquilado espacio para dos de las 350 dólares en stands de diez pies cuadrados y de alguna manera se las arregló para conseguir el espacio privilegiado cerca de la entrada de la sala de exposiciones. La idea era aprovechar ese descanso para presentar oficialmente el Apple II en la Feria. Aunque muchos en el Homebrew Club no consideraban que Apple fuera una entrada seria en el mercado (Gordon French vino un día y se fue burlándose de que la empresa seguía siendo básicamente dos tipos en un garaje), ahora había mucho dinero detrás de Apple. Un día, el nuevo presidente, Mike Scott, le había dicho a Chris Espinosa que copiara el software de demostración que ejecutaba un juego Breakout. Era un juego que Jobs había hecho para Atari y Woz había reescrito para Apple BASIC, y al final del juego, el programa calificaba tu puntuación con un comentario. Scott dijo, por cierto, ¿podría Chris cambiar también los comentarios, haciendo que la pantalla dijera "No es bueno" en lugar de "Pura mierda"? La razón era que algunos empleados del Bank of America venían a hablar sobre una línea de crédito.

Así que la gente de Apple estaba dispuesta a gastar para la feria. Contrataron a un decorador para que diseñara el stand y prepararon carteles de aspecto profesional con su nuevo y elegante logotipo, una manzana de colores del arco iris mordida. Trabajaron frenéticamente hasta el último minuto antes de tener que llevar las máquinas a San Francisco; habían planeado tener cuatro Apple II en

funcionamiento, y esos serían los únicos prototipos existentes. La noche del 15 de abril, llegaron las carcasas, recién hechas con moldes de inyección. Mientras todos trabajaban para colocar las entrañas de los ordenadores en esas carcasas, quedó claro lo diferente que era el Apple II de la competencia (con la posible excepción de la Sol). Las computadoras de los demás parecían algo como lo que un radiooperador de combate se hubiese colgado de la espalda. La Apple no tenía tornillos ni pernos visibles (los diez tornillos se enganchaban principalmente desde abajo): sólo una variación elegante, cálida y amigable de una máquina de escribir, futurista en su baja inclinación, pero no tan abruptamente inclinada como para parecer amenazante. Dentro de la máquina estaba la evidencia de la manipulación hacker de Woz. Había logrado reducir el número de chips a la asombrosa cantidad de sesenta y dos, incluida la poderosa unidad central de procesamiento 6502. De hecho, cuando abrías la tapa de la máquina, lo que veías era la "placa madre" de Woz (la tarjeta de circuito verde cargada de chips que era la Apple I, vitaminizada), una fuente de alimentación plateada del tamaño de un paquete de galletas Ritz y las ocho ranuras de expansión que indicaban los infinitos usos a los que podías aplicar la máquina. Tras insertarse los tornillos en los agujeros de los remaches de la carcasa, se colocaron las placas madre, se las atornilló, se probó todo y se colocaron las tapas, eran la una de la madrugada del debut mundial oficial de Apple.

A la hora indicada, las Apple estaban en el stand, cerca de la entrada. Casi todas las demás empresas confiaban en el telón de fondo de cortinas amarillas de eficacia probada con carteles de cartón pegados que deletreaban el nombre de la empresa en letras mayúsculas. Pero el stand de Apple brillaba con su logotipo de plexiglás de seis colores.

Jim Warren estaba en el lugar muy temprano esa mañana, por supuesto, con adrenalina después de sus dieciséis horas ininterrumpidas de preparación. Apenas dos días antes, él y Reiling habían constituido la Feria como una organización con fines de lucro. Aunque lo consideraba un "montón de tonterías burocráticas y legalistas", Reiling había señalado que, como sociedad, eran individualmente responsables de cualquier daño, y Warren había estado de acuerdo. En aquel momento, Jim Warren ya no tenía ninguna duda de hacia dónde se dirigía. Como persona que conocía bien la ética hacker, también podía ver lo que estaba sucediendo en su propio patio trasero de Silicon Gulch. El mundo real había llegado y era hora de una fusión entre las dos culturas, la hacker y la industrial, porque si se producía un choque, no habría dudas sobre quién perdería. Los hackers de hardware habían dejado salir a la luz el secreto de los microordenadores y los ingresos anuales multimillonarios de MITS, Processor Technology e IMSAI en 1976 eran una prueba irrefutable de que se trataba de una industria en crecimiento, digna de grandes sumas de dinero y de los cambios que ello implicaba. Jim Warren amaba el espíritu hacker, pero también era un superviviente. Si perdía dinero o sufría algún tipo de desastre por aferrarse a sus fobias posthippies, idealistas y antiburocráticas, no ayudaría en nada al hackerismo, mientras que ganar dinero tal vez no perjudicaría en absoluto a la ética hacker. Así que, aunque, como dijo más tarde, “no le importaban un carajo las cabinas, la electricidad, los contratos y todo eso”, siguió adelante. El micromundo estaba cambiando. No necesitaba más pruebas de ello que la escena en las taquillas del gran edificio de columnas griegas que era el Centro Cívico de San Francisco.

En ese soleado y luminoso día de abril de 1977, había miles de personas de pie en cinco largas colas, que serpenteaban por ambos lados del auditorio de una manzana de largo y se reunían en la parte de atrás. Un collar de cuentas de una manzana de largo de hackers, aspirantes a hackers, gente curiosa sobre hackers o gente que quería saber qué estaba pasando en este extraño mundo nuevo donde las computadoras significaban algo diferente a un tipo con camisa blanca y corbata negra y

billetera abultada y expresión apagada que, en conjunto, se sumaban a IBM. Es cierto que las colas estaban allí en gran parte porque la inexperiencia de Jim Warren había dado como resultado un desastre total en la preinscripción y la venta de entradas. Por ejemplo, en lugar de un precio fijo para la entrada el día de la venta, había diferentes tarifas: ocho dólares para el público en general, cuatro dólares para estudiantes, cinco dólares para los miembros del Homebrew Computer Club, etc. Y como los cajeros costaban diez dólares la hora, Warren había decidido no contratar demasiados extras. Ahora, con casi el doble de gente llegando de lo previsto, y todo el mundo parecía haber llegado temprano, era el tipo de situación que podía salirse de control.

Pero no se salió de control. Todo el mundo miraba a su alrededor con incredulidad de que toda esta gente estuviera interesada en las computadoras, de que la lujuria secreta de los hackers por las máquinas que habían tenido, a menudo cuando eran niños solipsistas, pequeños Greenblatts o Wozniaks, no fuera tan aberrante después de todo. Amar las computadoras ya no era una práctica pública prohibida. Así que no fue una pesadilla en absoluto estar con esta gente esperando para entrar a la Primera Feria Anual de Computadoras de la Costa Oeste. Como Jim Warren recordó más tarde: "Teníamos estas colas corriendo por todo el maldito edificio y nadie se irritó. Nadie fue insistente. No sabíamos lo que estábamos haciendo y los expositores no sabían lo que estaban haciendo y los asistentes no sabían lo que estaba pasando, pero todos estaban entusiasmados, eran agradables y poco exigentes y fue una experiencia tremendamente excitante. La gente simplemente se quedó parada y los expositores hablaban: '¿Tienes un Altair? ¡Eso es genial!' '¿Has resuelto este problema?' Y nadie se enfadaba".

Cuando la gente entraba en el salón, todo era tecnofanía, los sonidos de las voces se mezclaban con el traqueteo de las impresoras y los tonos metálicos de tres o cuatro estilos diferentes de música generada por ordenador. Si querías ir de un sitio a otro, tenías que medir qué parte del flujo constante de gente se movía en qué dirección, y abrirte paso a empujones hasta la corriente adecuada y seguirla hasta llegar a tu destino. Casi todos los expositores tenían stands llenos. En particular, Processor Technology, que estaba ejecutando el juego Target de Steve Dompier en ordenadores Sol. La gente también se apiñaba en el stand de IMSAI para hacerse un gráfico de biorritmos. Y justo allí, en la entrada, la ola del futuro, estaba Apple, ejecutando un programa de gráficos de vídeo caleidoscópico en un enorme monitor Advent. "Fue una locura", recordó más tarde Randy Wigginton, que trabajaba en el stand con Woz, Chris Espinosa y los demás. "Todo el mundo se acercaba y pedía demostraciones, y fue divertido porque la gente estaba entusiasmada".

La gente no sólo estaba entusiasmada con Apple. Era el triunfo de los hackers de hardware al convertir su pasión en una industria. Se podía ver la emoción cuando la gente miraba a su alrededor con incredulidad ante la gran cantidad de gente (¿toda esa gente?) y hubo un gran rugido cuando Jim Warren se puso en contacto con el sistema de megafonía y anunció la asistencia: el total del fin de semana fue de casi trece mil. Inmediatamente le siguió el autor de Computer Lib, Ted Nelson, sintiéndose sin duda como un gurú que alguna vez estuvo solo y que de un solo golpe se unió a un mar de discípulos. "Aquí el capitán Kirk", dijo Nelson. "¡Prepárense para el despegue!".

El propio Warren ya había pasado mucho tiempo desde el despegue. Dio una vuelta por la Feria en un par de patines, maravillándose de lo lejos que había llegado el movimiento. Para él, como para la gente de Apple, Processor Technology y docenas de otros lugares, este éxito tuvo implicaciones financieras muy bienvenidas; poco después de que terminara la Feria, después de recuperarse de un período de lo que más tarde llamaría "colapso extático", Warren estaría considerando si invertir sus ganancias en un Mercedes SL. Finalmente decidió comprar cuarenta acres de tierra que codiciaba

en las colinas con vistas a Woodside, y en pocos años habría construido una enorme estructura de madera con una terraza de secuoya y un jacuzzi con vista al Pacífico; sería su hogar y su lugar de trabajo informatizado, desde el cual un personal de más de una docena prepararía un pequeño imperio de publicaciones y exposiciones de computadoras. Jim Warren entendió el futuro.

La primera Feria de Computadoras fue para los hackers de hardware un evento comparable a Woodstock en el movimiento de los años sesenta. Al igual que el concierto en la granja de Max Yasgur, esto fue a la vez una reivindicación cultural y una señal de que el movimiento había crecido tanto que ya no pertenecía a sus progenitores. La última revelación tardó en asimilarse. Todo el mundo volaba, se desplazaba de un stand a otro, veía todo tipo de hardware revolucionario y software alucinante, conocía a gente con la que podía intercambiar subrutinas y esquemas de cableado, y asistía a algunos de los casi cien talleres, entre los que se encontraban Lee Felsenstein sobre el movimiento Community Memory, Tom Pittman sobre lenguajes informáticos, Bob Kahn sobre el programa informático del Lawrence Hall of Science, Marc LeBrun sobre música informática y Ted Nelson sobre el futuro triunfante.

Nelson fue uno de los oradores principales de un banquete celebrado en el cercano Hotel St. Francis. El título de su charla era “Esos inolvidables próximos dos años” y, mirando a esa masa de gente atraída por los micros, empezó diciendo: “Aquí estamos al borde de un nuevo mundo. Los pequeños ordenadores están a punto de rehacer nuestra sociedad, y ustedes lo saben”. En lo que a Nelson respectaba, la batalla estaba ganada: los hackers habían derrocado al malvado Profeta. “IBM estará en desorden”, cacareó Nelson. Era verdaderamente un mundo maravilloso el que estaba por desplegarse:

> Por ahora, sin embargo, las pequeñas computadoras están haciendo magia suficiente. Provocarán cambios en la sociedad tan radicales como los que trajeron el teléfono o el automóvil. Las pequeñas computadoras ya están aquí, se pueden comprar con tarjeta de crédito y los accesorios disponibles incluyen almacenamiento en disco, pantallas gráficas, juegos interactivos, tortugas programables que dibujan imágenes en papel de estraza y Dios sabe qué más. Aquí tenemos todos los ingredientes de una moda, que está floreciendo rápidamente hasta convertirse en un culto y pronto madurará hasta convertirse en un mercado de consumo en toda regla.
>
> ¡MODA! ¡CULTO! ¡MERCADO DE CONSUMO! La fiebre comenzará. La máquina publicitaria de la industria manufacturera estadounidense se volverá loca. La sociedad estadounidense se volverá loca. Y los próximos dos años serán inolvidables.

Capítulo 13. Secretos

El discurso de Ted Nelson no fue el estallido enloquecido de un planificador con una sobredosis de integración a gran escala. Los inolvidables dos años siguientes estuvieron marcados por un crecimiento sin precedentes en la industria que fue iniciada casi sin darse cuenta por los hackers de hardware. Los hackers del Homebrew o bien se dedicaron al negocio, se fueron a alguna de las nuevas compañías que se formaron en las etapas iniciales de este auge de los microordenadores, o siguieron haciendo lo que siempre habían estado haciendo: hackear. Los planificadores, aquellos que habían visto la llegada de los pequeños ordenadores como un medio para difundir el espíritu hacker, por lo general no se detuvieron a evaluar la situación: las cosas se movían demasiado rápido para contemplarlas. Dejaron a un lado a puristas como Fred Moore, quien una vez escribió en un tratado titulado “Pon tu confianza en la gente, no en el dinero” que el dinero era “obsoleto, sin valor, antivida”. El dinero era el medio por el cual el poder de los ordenadores estaba empezando a difundirse, y los hackers que ignoraban ese hecho estaban destinados a trabajar en un solipsismo (quizás feliz), ya sea en comunidades cerradas financiadas por la ARPA o en colectivos exiguos donde el término “al día” era una analogía clara para una existencia “de chip a máquina”.

La Feria de la Computadora de la Costa Oeste había sido el primer paso rotundo de los hackers de hardware que hacían su transición desde los garajes de Silicon Valley a los dormitorios y guaridas de Estados Unidos. Antes de finales de 1977, cayó el otro zapato. Empresas multimillonarias introdujeron combinaciones de terminal de computadora que no requerían ensamblaje, computadoras que se podían vender como electrodomésticos. Una de esas máquinas fue la Commodore PET, diseñada por el hombre que ideó el mismo chip 6502 que era el núcleo de Apple. Otra fue la computadora Radio Shack TRS-80, una computadora estampada en plástico, en línea de ensamblaje y vendida en masa en cientos de tiendas Radio Shack en todo el país.

Ya no era una lucha, un proceso de aprendizaje, fabricar computadoras. De modo que los pioneros del Homebrew, muchos de los cuales habían pasado de construir computadoras a fabricarlas, no tenían un vínculo común, sino la competencia para mantener la participación en el mercado. Esto retrasó la práctica tradicional del Homebrew de compartir todas las técnicas, de negarse a reconocer secretos y de mantener la información en un flujo sin trabas. Cuando se estaba considerando el Altair BASIC de Bill Gates, era fácil mantener la ética hacker. Ahora, como accionistas importantes de empresas que apoyaban a cientos de empleados, los hackers descubrieron que las cosas no eran tan sencillas. De repente, tenían secretos que guardar.

“Fue asombroso ver a los anarquistas ponerse una camiseta diferente”, recordó Dan Sokol más tarde. “La gente dejó de venir. Homebrew [todavía moderado por Lee Felsenstein, quien mantuvo viva la llama hacker] todavía era anarquista: la gente te preguntaba sobre la empresa y tenías que decir: 'No puedo decírtelo'. Yo lo resolví como lo hacían otras personas: no iba. No quería ir y no decirle cosas a la gente. No habría una salida fácil con la que te sintieras bien al respecto”.

El Homebrew todavía atraía a cientos de personas a sus reuniones, y su lista de correo superaba las mil quinientas, pero había muchos novatos allí, con problemas que no eran un desafío para los veteranos que habían construido máquinas cuando era casi imposible construirlas. Ya no era imprescindible ir a reuniones. Muchas de las personas implicadas en empresas como Apple, Processor Tech y Cromemco estaban demasiado ocupadas. Y las propias empresas proporcionaban las comunidades en torno a las cuales compartir información.

Apple era un buen ejemplo. Steve Wozniak y sus dos jóvenes amigos, Espinosa y Wigginton, estaban demasiado ocupados con la joven empresa como para seguir yendo a Homebrew. Chris Espinosa explicó más tarde: “[Después de la Feria de la Computación] nuestra asistencia a Homebrew empezó a disminuir y se acabó por completo a finales del verano de 1977. De hecho, creamos nuestro propio club de informática [en Apple] que estaba más centrado, más dedicado a producir cosas. Cuando empezamos a involucrarnos con Apple, descubrimos en qué queríamos trabajar y queríamos pasar todo nuestro tiempo perfeccionándolo, ampliándolo, haciendo más por ello, y queríamos profundizar en un tema en lugar de cubrir el campo y averiguar lo que hacía todo el mundo. Y así es como se hace una empresa”.

En cierto sentido, el “club de computadoras” en la sede de Apple en Cupertino reflejaba el mismo sentimiento de comunidad y de compartir del Homebrew. Los objetivos formales de la compañía eran tradicionales (ganar dinero, crecer, ganar participación en el mercado) y se requería cierto secretismo incluso de Steve Wozniak, quien consideraba que la apertura era el principio central de la Ética Hacker a la que se adhirió fervientemente. Pero esto significaba que la gente dentro de la compañía podía ser aún más cercana. Tenían que depender unos de otros para intercambiar sugerencias sobre BASIC de punto flotante o tarjetas de impresora paralelas. Y a veces, la comunidad era lo suficientemente flexible como para aceptar a algunos viejos amigos del Homebrew. Por ejemplo, a mediados de 1977, apareció John Draper.

El ex “Capitán Crunch” estaba en una mala situación. Al parecer, ciertas autoridades habían objetado su disposición a compartir secretos de la compañía telefónica con cualquiera que se molestara en preguntar; los agentes del FBI lo siguieron y, según sus relatos del incidente, colocaron un informante que lo convenció de una escapada a la caja azul mientras los agentes esperaban para arrestarlo. En esta segunda condena, fue sentenciado a una breve pena de prisión, y el encarcelamiento no le sentó bien al normalmente polémico Capitán, una persona acostumbrada a gritar como una hiena de dos metros de altura si alguien encendía un cigarrillo a seis pies de distancia de él. Después de su liberación, necesitaba urgentemente un trabajo legítimo, y Woz lo contrató como consultor, diseñando una placa de interfaz telefónica, algo que se conectaría a una de las ranuras de expansión de Apple para permitirle conectar el teléfono a su computadora.

Draper trabajó felizmente en la placa. La gente de Apple se divertía con su estilo de programación, que alternaba estallidos de brillantez con extraños desvíos pedantes. Draper era un programador "defensivo". Chris Espinosa, que tenía la poco envidiable tarea de intentar vigilar al impredecible Capitán, explicaría más tarde: "Digamos que estás escribiendo un programa y descubres que has hecho algo mal, como que cada vez que intentas usar el programa, aparece un botón. La mayoría de los programadores analizan su programa, descubren qué hace que salte el botón y lo solucionan para que no vuelva a hacerlo. Draper entraba y creaba un código para evitar el botón, de modo que cuando se produjera el error, el programa supiera que había cometido un error y lo solucionara, en lugar de evitar el error en primer lugar. El chiste es que, si Draper estuviera escribiendo rutinas matemáticas para la suma y se le ocurriera la respuesta 2 + 2 = 5, pondría una cláusula en el programa que dijera que si 2 + 2 = 5, entonces la respuesta sería 4. Esa es generalmente la forma en que escribe programas”.

Pero mientras los hackers de Apple se divertían con que el extraño estilo de John Draper estuviera dando como resultado un producto con muchas características, las personas a cargo de la parte comercial de Apple se enteraron de las capacidades del diseño de Draper. No les gustó. Apple no era un escaparate de trucos; esto no era Homebrew. Y la junta directiva de John Draper podía hacer

algunos trucos bastante ingeniosos; No sólo interactuaba con el teléfono, sino que generaba tonos oficiales de la compañía telefónica: era una caja azul controlada por computadora. Lo que Stew Nelson había hecho con el PDP-1 hace más de una década ahora se podía hacer en casa. El instinto hacker habría sido explorar las capacidades de este hardware, lo que le permitiría explorar sistemas en todo el mundo. Pero aunque Apple sintió que podría beneficiarse de la Ética Hacker al distribuir información sobre las entrañas de la máquina y distribuir sus computadoras como sistemas completos para explorar, no estaba en el negocio de promover el hackerismo puro. Era, después de todo, un negocio, con una línea de crédito y un camión lleno de capital de riesgo proporcionado por hombres con trajes de tres piezas que no se relacionaban con conceptos como el phreaking. "Cuando Mike Scott descubrió lo que [la placa de Draper] podía hacer", dijo Espinosa más tarde, "descartó el proyecto al instante. Era demasiado peligroso para ponerlo en el mundo para que alguien lo tuviera".

El abandono de ese proyecto se ajustaba perfectamente a la actitud de la floreciente Apple Computer Company, que vendía ordenadores como loca y se volvía respetable a un ritmo que dejó atónitos a los antiguos alumnos de Homebrew. Randy Wigginton, por ejemplo, se dio cuenta a finales del verano de 1977 de que esta empresa había eclipsado con creces la historia de crecimiento habitual. Fue entonces cuando todo el mundo fue a casa de Mike Markkula para celebrar el envío de equipos por valor de un cuarto de millón de dólares ese mes. Fue sólo el principio de una escalada que convirtió a Apple en una empresa de mil millones de dólares en cinco años.

Durante este período, cuando todo el mundo en Apple celebraba el aumento de los ingresos (montones de dinero que harían a muchos de ellos tan ricos que estarían más allá de los millonarios, en la capa de ozono del modo Creso, donde la riqueza se cuenta en unidades de decenas de millones), John Draper estaba en casa, jugando con su Apple. Puso la placa completa en su Apple II. La conectó a la línea telefónica. Y lo preparó para que "escaneara" centrales telefónicas enteras, buscando tonos delatores que le informaran de que había un ordenador al otro lado de la línea. Un ordenador virgen en el que un hacker pudiera entrar y explorar. Había hackeado un programa mediante el cual el ordenador podía marcar por sí solo. "Parecía algo inocente", dijo más tarde. Por sí solo, el ordenador empezó a hacer ciento cincuenta llamadas por hora. Cada vez que descubría un ordenador al otro lado de la línea, la impresora de teletipo conectada a la máquina procesaba el número de teléfono. Después de nueve horas, John Draper tenía una copia impresa de todos los números de ordenador de una central telefónica completa de tres dígitos. "Simplemente los recopilaba", explicaría más tarde. El sistema también podía detectar números de servicio de WATS Extenders, con los que se podían hacer llamadas de larga distancia gratuitas. (Fue el sistema de John Draper el que más tarde se usaría como modelo para el robo de ordenadores por parte de un joven hacker en la película WarGames.)

Desafortunadamente, el sistema siempre vigilante que era la compañía telefónica había desarrollado un nuevo equipo de detección de phreaking. La producción sin precedentes de John Draper de más de veinte mil llamadas telefónicas en menos de una semana no solo indicó que algo andaba mal, sino que también agotó el suministro de papel en la impresora de la compañía telefónica que registraba tales irregularidades. John Draper se enfrentó a otra visita de las autoridades. Era su tercera condena, la primera utilizando una computadora hogareña. Un comienzo desfavorable para una nueva era de phreaking con computadoras personales.

• • • • • • • •

Algunos pensaban que el establecimiento de una industria de ordenadores personales de bajo coste significaba que la guerra estaba ganada. Creían que la proliferación generalizada de ordenadores y sus lecciones innatas de apertura e innovación creativa, por sí mismas, estimularían la ética hacker. Pero para Lee Felsenstein, la guerra apenas estaba empezando.

Su pasión devoradora era la resurrección de la Memoria Comunitaria. Todavía se aferraba al sueño cuya gloria había vislumbrado en el experimento de Leopold's Records. Tal vez fuera una ironía exquisita que el desarrollo de la industria de las pequeñas computadoras hubiera sido ayudado en parte por la introducción del módem Pennywhistle, la placa de video VDM y la computadora Sol, todas ellas piezas de la mítica Terminal Tom Swift, una máquina que sólo podía llegar a buen puerto en las terminales de acceso público de las sucursales de Community Memory. Una ironía, porque entre los colegas de Lee había un consenso cada vez mayor de que el otrora audaz concepto de Community Memory (y la propia Terminal Tom Swift) habían sido suplantados por la rápida aceptación de las computadoras hogareñas. Estaba bien desear que una terminal pública fuera el corazón de un centro de información que sería una "amalgama de bibliotecas filiales, salas de juegos, cafeterías, parques urbanos y oficinas de correos". Pero ¿por qué la gente saldría de casa para ir a una terminal CM cuando podía usar una computadora Apple, junto con una interfaz telefónica allí mismo, en casa, para comunicarse con cualquier base de datos del mundo?

La propia Terminal Tom Swift podría haber sido archivada, pero Lee todavía se aferraba a sus objetivos. La novela de ciencia ficción en la que era protagonista estaba tomando giros argumentales más audaces, confirmando que era, en efecto, una obra de gran envergadura. En los dos "años inolvidables" transcurridos desde la triunfante Feria de la Computación, había visto desmoronarse una empresa. Processor Technology había sufrido demasiado crecimiento y una gestión demasiado deficiente para sobrevivir. Durante todo el año 1977, los pedidos del Sol llegaron a un ritmo que superó la capacidad de la empresa para satisfacerlos. En ese año fiscal, Bob Marsh calculó más tarde que la empresa había vendido cinco millones y medio de dólares, quizás ocho mil máquinas. Se trasladó a una sede limpia de treinta y seis mil pies cuadrados al este del Área de la Bahía.

Pero aunque el futuro parecía brillante, con Bob Marsh y Gary Ingram calculando que si las ventas llegaban a quince o veinte millones venderían y se harían ricos, la empresa estaba condenada al fracaso por la falta de planificación y por no haber sabido hacer frente a la competencia de las máquinas nuevas, más baratas y elegantes, como la Apple, la PET y la TRS-80. Marsh dijo más tarde que la compañía estaba pensando en entrar en ese segmento inferior del mercado, pero que se sentía intimidada por el poder de las empresas competidoras que habían anunciado ordenadores completos en el rango de los 1.000 dólares o menos. Pensó que PT podría vender el Sol como un artículo más caro y de calidad, como los amplificadores MacIntosh en el negocio del audio. Pero la compañía perdió la oportunidad de ampliar su equipamiento de forma efectiva cuando su sistema de almacenamiento en disco demostró ser poco fiable. Y no pudo entregar el software para sus máquinas a tiempo. Se anunciaban los próximos productos en el boletín de PT, una publicación animada que mezclaba informes de errores con citas crípticas ("'No hay enanos judíos' - Lenny Bruce"). Meses después, los productos, ya fueran programas de software o periféricos de hardware, seguían sin estar disponibles. Cuando PT recibió una oferta para vender ordenadores Sol a través de una nueva cadena de tiendas de ordenadores llamada Computerland, Marsh e Ingram se negaron, desconfiados porque los propietarios de la cadena eran las mismas personas que dirigían la empresa

(también en apuros, que pronto estaría en quiebra) que fabricaba el ordenador IMSAI. En lugar de vender las Sol como terminales de ordenador en Computerland, allí se venderían los Apple.

“Da vergüenza pensar en lo Mickey Mouse que éramos a veces”, admitió Marsh más tarde. No había un plan de negocios. Las cosas no se entregaban a tiempo, no se otorgaba crédito a los clientes prioritarios y los constantes errores de PT en las entregas y la falta de profesionalidad con los proveedores le dieron a la empresa una reputación de arrogancia y codicia.

“Simplemente estábamos violando algunas de las leyes básicas de la naturaleza”, dijo Marsh más tarde. Cuando las ventas se estancaron, se acabó dinero para hacer funcionar la empresa. Por primera vez buscaron inversores. Adam Osborne, un tábano ya establecido de la joven industria, les presentó a personas dispuestas a invertir, pero Marsh y Gary Ingram no querían renunciar a un porcentaje sustancial de la empresa. “Codiciosos”, dijo Osborne más tarde. Algunos meses después, cuando la empresa estaba casi en quiebra, Marsh volvió a aceptar la oferta. Ya no estaba abierta.

“Podríamos haber sido Apple”, dijo Bob Marsh, años después. “Mucha gente dice que 1975 fue el año del Altair, 1976 fue el año del IMSAI y 1977 fue el año de las Sol. Las máquinas dominantes”. Pero al final de esos “dos años inolvidables”, las compañías dirigidas por ingenieros que fabricaban esas máquinas, máquinas disponibles en forma de kit y ensambladas, máquinas con las que a los hackers de hardware les encantaba jugar… habían desaparecido. Los pequeños ordenadores dominantes en el mercado eran las Apple, PET, TRS-80, en los que el acto de creación de hardware esencialmente lo habían hecho por ti. La gente compraba estas máquinas para hackear software.

Lee Felsenstein fue quizás el mayor beneficiario financiero en la corta historia de Processor Technology. Nunca había sido un empleado oficial y sus regalías por la Sol finalmente sumaron más de cien mil dólares. Nunca le pagaron los últimos doce mil dólares de regalías. La mayor parte del dinero lo destinó a la nueva encarnación de Community Memory - que había establecido una sede en una gran estructura tipo loft de dos niveles, similar a un granero - en una zona industrial de West Berkeley. Efrem Lipkin y Jude Milhon, del grupo original, se encontraban entre los miembros dedicados del nuevo CM Collective, todos los cuales prometieron trabajar muchas horas y con salarios de subsistencia para establecer de manera permanente el emocionante experimento en el que habían trabajado a principios de la década. Requería un trabajo extenso desarrollar un nuevo sistema; el colectivo decidió que la financiación podría venir, en parte, escribiendo productos de software para estos pequeños ordenadores.

Mientras tanto, Lee estaba en bancarrota. “Lo racional para mí habría sido cerrar mi negocio [de ingeniería] y conseguir un trabajo. Pero no lo hice”, dijo más tarde. En cambio, trabajó por casi nada, diseñando una versión sueca de la Sol. Sus energías se dividían entre eso, las reuniones de Community Memory, desesperadamente serias, y las reuniones mensuales de Homebrew, que todavía moderaba con orgullo. El club era famoso ahora que los microordenadores estaban siendo aclamados como la principal industria en crecimiento del país. Y el mejor ejemplo de esto fue Apple Computer, que en 1980, año de su salida a Bolsa, recaudó 139 millones de dólares, lo que hizo que Jobs y Wozniak tuvieran una fortuna conjunta de más de 300 millones de dólares. Modo Creso.

Ese fue el año en que Lee Felsenstein se encontró con Adam Osborne en la Feria de la Computación. La feria de Jim Warren era ahora un evento anual que atraía a cincuenta mil personas en un fin de semana. Osborne era un inglés esbelto, nacido en Bangkok, de unos cuarenta años, con un fino bigote castaño y una vanidad imperiosa que impulsó a la notoriedad su columna en revistas especializadas (titulada “From the Fountainhead”). Ex ingeniero, hizo una fortuna publicando libros

sobre microcomputadoras cuando nadie más lo hacía. A veces llevaba cajas de ellas a las reuniones de Homebrew y volvía a casa con las cajas vacías y fajos de billetes. Sus libros acabaron vendiéndose por cientos de miles, McGraw-Hill compró su editorial y ahora, “con el dinero quemándome el bolsillo”, como decía, estaba pensando en dedicarse a la fabricación de ordenadores.

La teoría de Osborne era que todos los productos actuales estaban demasiado orientados a los hackers. No creía que a la gente le interesara saber acerca de la magia que los hackers encontraban en las computadoras. No sentía simpatía por la gente que quería saber cómo funcionaban las cosas, la gente que quería explorar cosas, la gente que quería mejorar los sistemas que estudiaba y con los que soñaba. En opinión de Adam Osborne, no se ganaba nada difundiendo la ética hacker; las computadoras eran para aplicaciones simples, como el procesamiento de textos o el cálculo financiero. Su idea era proporcionar una computadora sin lujos que viniera con todo lo necesario para funcionar; Osborne pensaba que la gente era más feliz cuando se liberaba de las opciones que producían ansiedad, como qué programa de procesamiento de textos comprar. Sería barata y lo suficientemente pequeña para llevarla en un avión. Una Volkscomputer portátil. Le pidió a Lee Felsenstein que la diseñara. Como la máquina que quería solo tenía que ser "adecuada", diseñarla no debería ser una tarea demasiado difícil. "Cinco mil personas en la península podrían haberlo hecho", dijo Osborne más tarde. “Resulta que yo conocía a Lee”.

Así que, para el veinticinco por ciento de esta empresa aún no formada, Lee Felsenstein diseñó la máquina. Decidió interpretar el requisito de Osborne de que la máquina fuera “adecuada” como si significara que podía hacer su trabajo habitual de ingeniería de desguace, asegurándose de que el diseño fuera lo suficientemente sólido como para soportar componentes bien probados en una arquitectura que evitara trucos y desvíos. “Ser capaz de hacer un diseño que sea bueno y adecuado, que funcione bien, que se pueda construir y que sea barato y que no contenga nada sofisticado es un problema artístico”, dijo más tarde. “Tenía que estar lo suficientemente loco y arruinado [para intentarlo]”. Pero Lee sabía que podía cumplir los requisitos. Como de costumbre, había miedo en la ecuación: Lee tenía un miedo irracional, sin duda, a Adam Osborne; supuso que identificaba a Adam con las figuras de autoridad de su infancia. No había forma de que estos dos pudieran comunicarse profundamente. Una vez Lee intentó explicarle Community Memory (su verdadera carrera) y Osborne “no lo entendió”, se lamentó Lee. “Puede que sea uno de los últimos en entender lo que Community Memory es, cuando lo ves, lo usas”. Sin embargo, Lee trabajó duro para Adam Osborne, trabajando en un espacio en la sede de Community Memory, y en seis meses la abandonó.

Creía que había cumplido con los requisitos técnicos y artísticos al construir la máquina que se conoció como Osborne 1. Los críticos dirían más tarde que la máquina con carcasa de plástico tenía una pantalla de cinco pulgadas incómodamente pequeña y señalarían otros pequeños problemas, pero cuando la computadora salió a la venta por primera vez, los elogios fueron abundantes y Osborne Computer pronto se convirtió en una empresa multimillonaria. Y, de la nada, Lee Felsenstein valía más de veinte millones de dólares. Sobre el papel.

No cambió radicalmente su estilo de vida. Seguía viviendo en el espartano apartamento del segundo piso que alquilaba por menos de doscientos dólares al mes. Seguía lavando su ropa en lavanderías poco iluminadas cerca de las oficinas de Osborne en Hayward. La única concesión era que conducía un coche de empresa, un BMW nuevo. Pero quizá debido a la edad, algunas sesiones de terapia y la madurez, así como a su éxito tangible, había crecido en otros aspectos. A finales de sus treinta, se describió a sí mismo como “todavía poniéndose al día, pasando por experiencias que uno

normalmente vive cuando tiene poco más de veinte años". Tenía una novia estable, una mujer que trabajaba en Osborne.

De las acciones de Osborne que Lee vendió, casi todas fueron a Community Memory, que, en medio del auge de los microordenadores, estaba pasando por momentos difíciles.

Gran parte de las energías del colectivo se destinaban a desarrollar software para vender y ganar dinero para el establecimiento del sistema sin fines de lucro Community Memory. Pero dentro del grupo se estaba debatiendo intensamente sobre la conveniencia de vender el software a cualquiera que quisiera usarlo o restringirlo para que no beneficiara a ningún esfuerzo militar. No estaba claro que los militares estuvieran clamando por comprar este software, que incluía una base de datos y aplicaciones de comunicaciones más útiles para las pequeñas empresas que para los portadores de armas. Pero estos eran radicales empedernidos de Berkeley, y era de esperar que se produjeran discusiones como estas. La persona que más se preocupaba por el ejército era Efrem Lipkin, el hacker bendecido con la magia de la informática y maldecido con un odio hacia el uso que se daba a las computadoras.

Lee y Efrem no se llevaban bien. A Efrem no le gustaba la industria de las computadoras personales, que consideraba "juguetes de lujo para la clase media". Consideraba que la computadora de Osborne era "repugnante". Le molestaba que Lee trabajara para Osborne mientras él y los demás trabajaban por estipendios de esclavos en CM. El hecho de que gran parte del dinero para CM viniera del trabajo de Lee en esa máquina molestaba a Lipkin como un error en un programa, un error fatal que no se podía eliminar con un código. Lipkin era un purista de los hackers; aunque él y Lee estaban de acuerdo en el espíritu de la Memoria Comunitaria (utilizar computadoras para unir a la gente), no podía aceptar ciertas cosas. Efrem Lipkin le dijo al grupo que una cosa que no podía aceptar era ninguna venta del software que había escrito a los militares.

El problema era más profundo que eso. Los ordenadores personales como el Apple y el Osborne, junto con los módems al estilo del Pennywhistle de Lee, habían engendrado otros ejemplos del tipo de cosas que Community Memory estaba intentando hacer. La gente utilizaba los ordenadores para comunicarse. Y el mito original de Community Memory, el ideal de máquinas de gracia amorosa en un campo que nos vigilaba, se había cumplido en gran medida: en menos de diez años, los ordenadores habían sido desmitificados. Ya no eran cajas negras malvadas a las que temer. Incluso estaban de moda: con el tiempo, la tecnología informática no sólo sería algo común en Leopold's Records, sino que probablemente se vendería allí, en forma de software que reemplazaría a los discos en algunos de los estantes. Jude Milhon, amiga íntima tanto de Lee como de Efrem, una persona que había dedicado una parte sustancial de su vida a Community Memory, apenas podía pronunciar las palabras cuando hablaba de ello, pero lo sabía: la habían arruinado. La Rebelión de 2100 había terminado, y todavía no era 1984. Los ordenadores eran aceptados como herramientas de convivencia y su potencia era accesible en miles de tiendas minoristas para aquellos que podían pagarla.

Lleno de frustración, Efrem Lipkin estalló en una reunión. Expuso lo que consideraba el fracaso del grupo. "Básicamente, pensé que la cosa se estaba desmoronando", dijo más tarde. Fue especialmente duro con el tema del dinero de Lee que financiaba al grupo.

Lee le dijo que ese dinero contaminado estaba pagando el salario de Efrem.

"Ya no", dijo Efrem. Y el hacker se fue.

Menos de un año después, Osborne Computer ya no existía. Los errores de gestión peores que en Processor Technology habían convertido a la empresa en el primero de muchos desastres financieros importantes en lo que se llamaría “La Gran Sacudida Informática”. Los millones de Lee en papel se esfumarían.

Pero aún tendría sus sueños. Se había ganado una gran batalla. Ahora, tal vez en dos tercios de la épica novela de ciencia ficción, era hora de reunir fuerzas para un giro final hacia la grandeza. Algún tiempo antes de que Osborne Computer se declarara en quiebra, Lee había estado lamentando la naturaleza opaca de los ordenadores más recientes, de la falta de necesidad que llevaría a la gente a entrar en los chips y las placas de circuitos y cablearlos. La construcción de hardware, decía, es una forma objetivada de pensar. Sería una pena que eso se desvaneciera, que se limitara sólo a unos pocos. No creía que desapareciera. “[La magia] siempre estará ahí hasta cierto punto. Hablas de deus ex machina, bueno, estamos hablando de deus in machina. Empiezas pensando que hay un dios en la caja. Y luego descubres que no hay nada en la caja. Pones al dios en la caja”.

Lee Felsenstein y los hackers de hardware habían ayudado a hacer la transición del mundo de los hackers del MIT, donde la ética hacker sólo podía florecer dentro de las limitadas comunidades monásticas alrededor de la máquina, a un mundo donde las máquinas estaban en todas partes. Ahora se estaban fabricando millones de ordenadores, cada uno de ellos una invitación a programar, a explorar, a mitificar en lenguaje de máquina, a cambiar el mundo. Las computadoras salían de las líneas de montaje como pizarras en blanco; una nueva generación de hackers se sentiría seducida por el poder de llenar las pizarras; y el software que crearan sería presentado a un mundo que veía las computadoras de una manera muy diferente a la de una década antes.

Parte III. Hackers de juegos: Las Sierras: Los Ochentas

Capítulo 14. El mago y la princesa

Al salir de Fresno en dirección noreste por la Ruta 41 hacia la Puerta Sur de Yosemite, subías lentamente al principio, a través de campos bajos salpicados de enormes rocas picadas. A unas sesenta kilómetros estaba el pueblo de Coarsegold; poco después, la carretera ascendía abruptamente, hasta la cima de una montaña llamada Deadwood. Solo después de comenzar el descenso desde Deadwood se podía ver cómo la Ruta 41 formaba la franja central de Oakhurst. Población inferior a seis mil habitantes. Un moderno polimercado llamado Raley's (desde alimentos saludables hasta mantas eléctricas). Unos cuantos locales de comida rápida, varios grupos de tiendas especializadas, dos moteles y una oficina inmobiliaria con una estatua de fibra de vidrio marrón descolorida de un oso en el exterior. Después de una milla o así de Oakhurst, la carretera continuaba su ascenso hacia Yosemite, a treinta millas de distancia.

El oso podía hablar. Pulsaba un botón en su base y recibía una bienvenida grave y gruñona a Oakhurst, un discurso sobre el precio del terreno. El oso no mencionaba la transformación de la ciudad gracias al ordenador personal. Oakhurst había pasado por tiempos difíciles, pero en 1982 se jactaba de una historia de éxito importante. Una empresa construida, en cierto sentido, por el sueño hacker, y hecha posible sólo por la magia de Steve Wozniak y su Apple Computer. Una empresa que simbolizaba cómo los productos de los hackers (programas informáticos que son obras de arte) habían sido reconocidos como tales en sectores importantes del mundo real. Los hackers que jugaban a Spacewar en el MIT no lo imaginaron, pero el descendiente de ese programa PDP-1, ahora que los hackers de hardware habían liberado el ordenador y lo habían convertido en algo personal, había engendrado una nueva industria.

No muy lejos de Talking Bear había un discreto edificio de dos pisos construido para oficinas y tiendas. A excepción de un pequeño salón de belleza, un despacho de abogados y la diminuta oficina local de Pacific Gas and Electric, todo el edificio estaba ocupado por la empresa Sierra On-Line. Su producto principal era el código, líneas de código informático en lenguaje ensamblador escritas en disquetes que, al insertarse en ordenadores personales como el Apple, se convertían mágicamente en fantásticos juegos. Una especialidad de la empresa eran los juegos de "aventura", como el que perfeccionó Don Woods en el laboratorio de inteligencia artificial de Stanford; esta empresa había descubierto cómo añadir imágenes al juego. Vendió decenas de miles de estos discos.

En ese día de agosto de 1982, On-Line tenía alrededor de setenta empleados. Las cosas cambiaron tan rápidamente que en un día determinado era difícil dar una cifra exacta, pero era más del triple de los empleados que tenía un año antes. Un año antes, sólo estaban los dos fundadores, Ken y Roberta Williams, que tenían, respectivamente, veinticinco y veintiséis años cuando fundaron la empresa en 1980.

Ken Williams estaba sentado en su oficina. Afuera estaba su Porsche 928 rojo. Era otro día para hacer historia y divertirse un poco. La oficina de Ken hoy estaba relativamente ordenada; las pilas de papeles sobre el escritorio tenían solo unos pocos centímetros de alto, el sofá y las sillas frente al escritorio estaban libres de disquetes y revistas. En la pared había una litografía, homenaje al Pensador de Rodin: en lugar de ese noble humano congelado en su lúcida mente, había una representación de un robot contemplando una Apple de colores del arco iris.

Ken Williams, por su parte, estaba característicamente desaliñado. Era un hombre corpulento y de gran barriga, con rasgos hinchados que abrumaban sus amigables ojos azules. Había un agujero en su camiseta roja y un agujero en sus jeans. Su cabello rubio oscuro hasta los hombros cubría su

cabeza en una maraña despeinada. Estaba sentado en su alto sillón ejecutivo marrón como un King Cole poscontracultural. En una agradable cadencia californiana puntuada por comentarios modestos que salían de su lengua con nostalgia, le estaba explicando su vida a un periodista. Había cubierto el tremendo crecimiento de su empresa, su placer en difundir el evangelio de las computadoras al mundo a través del software que vendía su empresa, y ahora estaba hablando de los cambios que se habían producido cuando la empresa se hizo grande, algo mucho más que una operación de hackers en las colinas. Ahora estaba en contacto con el poder del mundo real.

"Las cosas que hago a diario me dejan atónito", dijo.

Habló de salir a bolsa con el tiempo. En 1982, mucha gente que poseía empresas engendradas por la revolución que habían iniciado los hackers de hardware hablaba de esto. Las computadoras se habían convertido en la joya de la economía, la única área de crecimiento real en un período de recesión. Cada vez más gente veía la magia vislumbrada por primera vez en los monasterios procesados por lotes por los visionarios prácticos; en el poder aprovechado por los artistas del PDP-1; en el dominio accesible de la información proporcionada por Ed Roberts y promovida por Lee Felsenstein. Como resultado, compañías como Sierra On-Line, que comenzaron con muy pocos recursos, ahora eran lo suficientemente grandes como para contemplar la posibilidad de ofrecer acciones en bolsa. La charla de Ken Williams recordaba a la que se escuchó varios años antes, cuando, utilizando las mismas cadencias conscientemente despreocupadas, la gente hablaba de que un día iban a ser engañados: en ambas circunstancias, un acto que antes se abordaba con gravedad evangelizadora ahora se consideraba una deliciosa inevitabilidad. Salir a bolsa era algo que uno consideraba naturalmente, al menos cuando había pasado de ser un ambicioso programador informático a ser propietario de una empresa de juegos de ordenador que facturaba 10 millones de dólares al año en poco más de dos años.

Era un momento crucial para la empresa de Ken Williams. También era un momento crucial para la industria de los juegos de ordenador, un momento crucial para la industria informática en su conjunto y un momento crucial para Estados Unidos. Los elementos habían conspirado para poner a Ken Williams, un autodenominado ex hacker, en el asiento del conductor de algo más que un Porsche 928.

Ken Williams salió de su oficina y se dirigió a una gran sala dos puertas más abajo en el mismo edificio. Había dos filas de cubículos en esta sala con paredes de yeso y alfombra industrial. En cada cubículo había un pequeño ordenador y un monitor. Esta era la oficina de programación, y aquí era donde un joven hacker había venido a mostrarle su juego a Ken Williams. El hacker era un chico de aspecto arrogante; Era bajito, tenía una sonrisa de bravucón en una cara de nariz respingada y su pecho sobresalía como el de un gallo bajo una camiseta azul descolorida. Había llegado en coche desde Los Ángeles esa mañana, tan drogado que podría haber llenado el depósito con su exceso de adrenalina.

En el monitor había un prototipo de un juego llamado Wall Wars, escrito en los últimos meses en ráfagas intensas entre la medianoche y las ocho de la mañana. Mientras el hacker había trabajado en un pequeño apartamento, su estéreo había puesto a todo volumen música de Haircut 100. Wall Wars implicaba una serie de piezas de colores, similares a ladrillos, que formaban una pared cinética en el centro de la pantalla. En la parte superior e inferior de la pantalla había criaturas parecidas a robots igualmente deslumbrantes. Un jugador controlaría uno de los robots, atravesaría la pared derribando

suficientes ladrillos para formar un hueco móvil y destruiría al otro robot, que, por supuesto, estaría intentando realizar la misma tarea, con el jugador como víctima.

El hacker se había prometido a sí mismo que si Ken Williams compraba su concepto de juego, dejaría su trabajo como programador de Mattel y se independizaría, uniéndose a las filas de un grupo de élite al que ya se hacía referencia como Superestrellas del Software. Eran el apogeo de una Tercera Generación de hackers que habían aprendido su arte de programación en pequeñas computadoras, que nunca se habían puesto en marcha a través de una comunidad. Que soñaban no solo con el hack definitivo, sino con la fama y grandes cheques de regalías.

Ken Williams entró en la habitación y apoyó un codo en el borde del cubículo. El joven hacker, enmascarando su nerviosismo, comenzó a explicar algo sobre el juego, pero Ken no parecía escuchar.

“¿Esto es todo hasta ahora?” dijo Ken.

El hacker asintió y comenzó a explicar cómo se jugaría el juego. Ken lo interrumpió.

“¿Cuánto tiempo te llevará terminarlo?”

“Voy a dejar mi trabajo”, dijo el hacker. “Puedo hacerlo en un mes”.

“Calculamos que en dos meses”, dijo Ken. “Los programadores siempre mienten”. Se dio la vuelta y comenzó a alejarse. “Pasa por mi oficina y te haremos firmar un contrato”.

Era como un magnate del entretenimiento de antaño dando el visto bueno a una estrella en audición. Era indicativo del cambio masivo en la forma en que la gente pensaba sobre las computadoras, las usaba e interactuaba con ellas. La historia de los hackers del MIT y el Homebrew Club había llevado a esto: Sierra On-Line y aspirantes a estrellas del software.

La ética hacker había llegado al mercado.

• • • • • • •

Ken Williams nunca fue un hacker puro. Ciertamente no tomó el apelativo como una insignia de orgullo; la idea de una aristocracia de excelencia informática nunca se le ocurrió. Había tropezado con la informática. Sólo de manera incidental desarrolló una relación con la máquina, y no fue hasta que se creyó su amo que empezó a apreciar qué tipo de cambios podía hacer la computadora en el mundo.

Al principio, la computadora lo tenía totalmente desconcertado. Fue en el campus de Pomona de la Politécnica de California, al que asistía Ken Williams porque (a) costaba sólo veinticuatro dólares el trimestre más los libros y (b) tenía sólo dieciséis años y estaba cerca de casa. Su especialidad era física; tenía problemas con las clases. Aunque Ken siempre había tenido un rendimiento académico bajo gracias a su gran aptitud, no dominaba con tanta facilidad materias como la trigonometría y el cálculo como las materias de la escuela secundaria. Ahora había un curso de computación, orientado a la programación en FORTRAN.

Ken Williams se sentía intimidado por las computadoras, y esa intimidación desencadenó en él una reacción extraña. Siempre se había resistido a los planes de estudio preestablecidos: aunque se negaba a hacer los deberes en la escuela secundaria, leía casi compulsivamente de todo, desde los Hardy Boys hasta lo que se convirtió en su género favorito, las historias de pobreza a riqueza de Harold Robbins. Se identificaba con los desvalidos. El padre de Williams era reparador de

televisores en Sears, un hombre rudo que se había mudado a California desde el condado de Cumberland, Kentucky; sus compañeros de trabajo lo apodaban "Country". Ken creció en un barrio bastante duro en Pomona, a veces compartiendo dormitorio con sus dos hermanos. Evitaba las peleas con asiduidad, admitiendo después alegremente que era “un cobarde”. “No le devolvería el golpe”, explicó una vez, como si los ritos de dominación y la postura machista le fueran ajenos.

Pero cuando leyó sobre esas luchas en grandes novelas melodramáticas, se sintió embelesado. Le encantaba la idea de un chico pobre que ganaba mucho dinero y se quedaba con todas las chicas. Era susceptible a los encantos hiperbólicos de una vida como la de Jonas Cord, el joven despiadado personaje parecido a Howard Hughes en The Carpetbaggers que convirtió su herencia en un imperio de aviación y cine. “Ahí es donde encontré mi modelo a seguir”, explicó Williams más tarde. Tal vez fue algo de la ambición de Jonas Cord lo que llevó a Ken Williams a volverse más activo en la escuela secundaria, donde se unió a la banda, tuvo una novia, aprendió a jugar al juego de las buenas notas y elaboró planes para ganar dinero. (Más tarde se jactaría de haber ganado tantos concursos de ventas en su ruta de reparto de periódicos que se conocía de tú a tú con los encargados de recoger las entradas en Disneylandia.) La tendencia de Ken a la autocrítica y su independencia aparentemente informal enmascaraban una determinación feroz que se manifestaba incluso cuando un ordenador de Control Data lo acorralaba en la clase de FORTRAN.

Durante semanas luchó, quedando rezagado respecto de sus compañeros de clase. Se había planteado un problema: simular un ratoncito que corría por un laberinto, seguía una pared y salía del laberinto. (Requería un programa similar al antiguo programa del Ratón en el Laberinto del TX-0, donde el ratoncito intentaba encontrar las copas de martini.) Transcurridas seis semanas del curso de nueve semanas, Ken iba camino de una F. Y no había nada que a Ken Williams, incluso entonces, le gustara del fracaso. Así que siguió intentándolo hasta que un día se dio cuenta de repente de que el ordenador en realidad no era tan inteligente. Era solo una bestia tonta que seguía órdenes y hacía lo que le decías exactamente en el orden que determinabas. Podías controlarla. Podías ser Dios.

> ¡Poder, poder, poder! Allí arriba, donde el mundo era como un juguete debajo de mí. Donde sostenía el palo como si fuera mi polla en mis manos y no había nadie... ¡que me dijera que no!
>
> --Jonas Cord, en The Carpetbaggers de Harold Robbins

El ratón atravesó el laberinto. Ken Williams pasó el curso. Fue como si se le hubiera encendido una luz en la cabeza, y todos en la clase pudieron verlo por la facilidad con la que descifró el código. Ken Williams tenía algo con la Bestia Tonta.

Una relación más importante para Ken en ese momento era su romance con una chica llamada Roberta Heuer. La había conocido en la escuela secundaria, cuando ella estaba saliendo con un amigo suyo. De repente, dos meses después de una cita doble, Ken la llamó, le recordó nerviosamente quién era y la invitó a salir. Roberta, una chica recatada y pasiva, dijo más tarde que al principio no le había impresionado mucho Ken. “Era lindo, pero pensé que actuaba como un tonto. Era tímido pero [para compensarlo] se excedía, actuando demasiado agresivo. Llevaba cigarrillos en el bolsillo, pero no fumaba. Me pidió que fuéramos novios la primera semana [que salimos]”.

Roberta había estado saliendo con un chico que vivía en el norte del estado. Ken intentó obligarla a elegir entre ellos. Roberta bien podría haber decidido no tener a ese chico inseguro y agresivo, pero un día Ken se sinceró con ella. “Estaba hablando de física”, recordó Roberta más tarde. “Pensé que realmente era un chico inteligente. Todos los novios que había tenido antes eran bastante tontos. Ken hablaba de cosas reales, de responsabilidad”. Dejó de salir con el otro chico y casi al instante Ken presionó para un compromiso permanente. “No quería estar sola”, reflexionó más tarde.

Roberta habló con su madre al respecto: “Se va a ir a algún lado”, dijo. “Para triunfar de verdad. Para ser alguien”.

Finalmente, Ken le dijo: “Nos casaremos y eso es todo”. Ella no se opuso. Tenía diecinueve años; él era un año más joven.

En menos de un año, Roberta estaba embarazada y Ken estaba sacando D y preocupándose por mantener a una familia. Sabía por los anuncios de empleo que había muchos más trabajos en programación informática que en física, así que pensó, tal como decían las tapas de las cajas de cerillas, que encontraría una carrera en el procesamiento electrónico de datos. El padre de Roberta firmó un préstamo estudiantil de 1.500 dólares para la matrícula de una escuela de oficios llamada Control Data Institute.

El mundo en el que Ken Williams estaba entrando no se parecía en nada al sagrado coto del laboratorio de inteligencia artificial del MIT. Sus posibles colegas en el campo de la informática empresarial tenían poco del hambre práctica que impulsaba a la clase de graduados de Altair que hackeaban hardware. A principios de los años 70, el campo de la informática empresarial en el que Ken estaba entrando se consideraba el más espeluznante de Estados Unidos. Era una broma, una ocupación en la que unos topos mansos hacían cosas (¿quién sabe qué eran esas cosas?) con las tarjetas perforadas y las ruedas giratorias de enormes ordenadores gigantes. En lo que respecta al público, ni siquiera había mucha diferencia entre los zánganos que perforaban mecánicamente las tarjetas y golpeaban los teclados y los técnicos expertos que programaban las máquinas para poner las tarjetas en su lugar. Todos eran vistos como los topos de camisa blanca y anteojos gruesos en la sala de ordenadores. Criaturas de la era incorpórea.

Si Ken y Roberta hubieran formado parte de un amplio círculo de amigos, tal vez hubieran tenido que enfrentarse a ese estereotipo, al que Ken no se parecía en lo más mínimo. Pero Ken y Roberta no se molestaron en echar raíces ni en establecer amistades estrechas. Como programador informático, Ken era menos un Richard Greenblatt o un Lee Felsenstein que un Jonas Cord. Más tarde, diría con desenfado: "Supongo que la avaricia me definiría mejor que cualquier otra cosa. Siempre quiero más".

Ken Williams estaba lejos de ser un programador deslumbrante cuando terminó el Control Data Institute, pero sin duda estaba preparado para hacer todo lo que se le pidiera. Y más. Tanto trabajo como fuera posible, para ayudarle a llegar lo más alto que pudiera. Después aceptar otro trabajo más exigente, estuviera o no cualificado. En lugar de romper con su anterior empleador, Ken intentó seguir en nómina, en modo consultor.

Afirmaba conocer lenguajes informáticos y sistemas operativos de los que no sabía nada, leía un libro sobre el tema horas antes de una entrevista de trabajo y se ganaba el puesto con mentiras. “Bueno, estamos buscando un programador en BAL”, le decían, refiriéndose a un lenguaje informático esotérico, y él se reía casi con desdén.

“¿BAL? ¡He estado programando en BAL durante tres años!”.

Después salía corriendo a buscar algunos libros, ya que nunca había oído hablar de BAL. Pero cuando empezaba el trabajo ya había conseguido documentación, enterrada uniformemente en densos manuales de hojas sueltas impresos a bajo precio, para simular ser experto en el “entorno BAL”, o al menos ganar tiempo hasta poder entrar en la máquina y adivinar los secretos de BAL.

No importaba dónde trabajara, en cualquier cantidad de empresas de servicios anónimas en el enorme valle sobre Los Ángeles, Ken Williams no conocía a una sola persona que mereciera un ápice de su respeto. Observaba a gente que llevaba años programando ordenadores y se decía a sí mismo: «Dadme un libro y en dos horas estaré haciendo lo que ellos hacen». Y, en efecto, después de montones de manuales y unos cuantos días de trabajo de catorce horas, al menos parecía un programador de primera.

Acudía a los santuarios informáticos, con un aire acondicionado muy potente, a horas intempestivas de la noche para corregir un error o para poner en marcha el ordenador cuando uno de sus programas se alimentaba a sí mismo por accidente y hacía que millones de cálculos se volvieran locos en tal arrebato de malentendidos que nada de lo que se le ocurriera al equipo habitual podía revivir la máquina. Pero Ken, seguro de que la estupidez de sus colegas sólo quedaba eclipsada por la asombrosa obediencia de la Bestia Estúpida a la que podía alimentar y con la que hacerse amigo gracias a sus habilidades de programación, trabajaba tres días seguidos, olvidándose incluso de parar a comer, hasta que la Bestia Estúpida volvía al trabajo. Ken Williams, el héroe del día, el domador de la Bestia Muda, se iba a casa, dormía un día y medio y luego volvía al trabajo, listo para otro maratón. Los empleadores se daban cuenta y lo recompensaban.

Ken ascendía a una velocidad cuántica: Roberta calculó que se habían mudado a varios lugares de la zona de Los Ángeles unas doce veces en esa década de auge, siempre asegurándose de obtener ganancias con la casa. No tenían tiempo para hacer amigos. Se sentían solitarios e inadaptados, normalmente la única familia de cuello blanco en un barrio de clase trabajadora. El consuelo era el dinero. “¿No sería bueno ganar doscientos dólares más a la semana?”, preguntaba Roberta, y Ken conseguía un nuevo trabajo o aceptaba más trabajos de consultoría... pero incluso antes de que Ken se hubiera adaptado a este nuevo trabajo, él y Roberta estaban sentados en la pequeña sala de estar de la casa en la que vivían y decían: “¿No sería bueno ganar doscientos dólares más?”. La presión nunca cesó, especialmente porque Ken Williams tenía sueños vanos de fantásticas sumas de dinero, dinero suficiente para holgazanear durante el resto de su vida; no solo todo el efectivo que él y Roberta pudieran gastar, sino también todo el que sus hijos pudieran gastar (Roberta estaba embarazada por entonces del segundo hijo de Williams, Chris). ¿No sería bueno, pensó, retirarse a los treinta?

Para entonces, algo más estaba cambiando: su relación con la Bestia Tonta. Cuando Ken tenía tiempo, solía sacar algunos de esos manuales densos y de hojas sueltas impresos a bajo precio, tratando de averiguar qué hacía que la gran máquina de Burroughs o IBM o Control Data funcionara realmente. A medida que ganó competencia en su profesión, comenzó a respetarla más; a ver cómo podía acercarse al arte. Había capas de experiencia que estaban mucho más allá de lo que Williams había llegado a suponer anteriormente. Existía un panteón de programación, casi como una especie de hermandad filosófica de la vieja escuela.

Ken había probado este reino más exótico cuando se las arregló para conseguir un trabajo como programador de sistemas para Bekins Moving and Storage. Bekins estaba pasando entonces de una

computadora Burroughs a una máquina IBM más grande y ligeramente más interactiva. Ken descaradamente se inventó una historia profesional de hechicería en IBM y consiguió el trabajo.

En Bekins, Ken Williams se enganchó a la programación pura. Su tarea era instalar un sistema de telecomunicaciones de alta potencia en IBM que permitiera a una computadora dar soporte a ochocientos o novecientos usuarios en el campo en todo el país, y los problemas y complicaciones superaban todo lo que había enfrentado hasta entonces. Experimentaba con tres o cuatro lenguajes que no tenían nada que ver con su trabajo, fascinado con las técnicas y los marcos mentales requeridos con cada lenguaje. Había todo un mundo dentro de esta computadora... una forma de pensar. Y tal vez por primera vez Ken Williams se sentía atraído por el proceso de computación más que por el objetivo de completar una tarea. En otras palabras, hackear.

Como consecuencia de su interés sostenido, Ken permaneció en Bekins más tiempo que en la mayoría de sus otros empleadores: un año y medio. Fue un tiempo bien empleado, ya que su siguiente trabajo le presentaba un desafío aún mayor, así como contactos e ideas que pronto le permitirían hacer realidad sus fantasías más salvajes.

• • • • • • • •

La empresa se llamaba Informatics. Fue una de las muchas firmas que surgieron a mediados de los años sesenta para aprovechar un vacío en el campo del software para mainframes. Cada vez más grandes empresas y agencias gubernamentales adquirían computadoras, y casi ninguno de los programas que suministraban las gigantescas empresas informáticas podía ejecutar con destreza las tareas que se suponía que debían realizar las computadoras. De modo que cada empresa tenía que contratar a su propio personal de programación o confiar en consultores altamente pagados que invariablemente desaparecían justo cuando el sistema fallaba y salían datos valiosos que parecían rusos. Entonces saldría un nuevo equipo de programadores o consultores para desenredar el embrollo, y el proceso se repetiría: empezando desde cero, el nuevo equipo tendría que reinventar la rueda.

Informatics y otras empresas similares se crearon para vender software que hiciera que los gigantes de Hulking fueran un poco más comprensibles. La idea era inventar la rueda de una vez por todas, patentarla y venderla como locos. Sus programadores trabajarían arduamente en el nivel de ensamblaje y finalmente darían con un sistema que permitiera a los programadores de bajo nivel, o incluso en algunos casos a los no programadores, realizar tareas informáticas simples. Después de todo, todos estos sistemas comerciales hacían más o menos lo mismo: recibías algo de un empleado o de una sucursal en papel que se tecleaba y se introducía en un sistema que modificaba algún archivo preexistente. La informática ideó un sistema preprogramado llamado Mark 4. En algún momento de los años setenta se convirtió en el producto de software para mainframes más vendido de todos los tiempos, llegando a alcanzar los 100 millones de dólares en ingresos anuales.

A finales de los años setenta, uno de los directivos a cargo de los nuevos productos de Informatics era Dick Sunderland, un ex programador de FORTRAN que estaba subiendo la escalera corporativa después de renunciar a regañadientes a un intento tardío de estudiar derecho. En lugar de estudiar derecho, Sunderland había decidido entablar un romance con un concepto brillante y sagrado de gestión. Ser un líder de hombres, un hábil constructor de equipos de empleados competentes y bien coordinados, un promotor persuasivo y un manipulador constructivo... eso era a lo que aspiraba Dick Sunderland.

Sunderland, un hombre pequeño, de tez pálida, ojos entornados y acento contemplativo, se consideraba un gerente natural. Siempre le había interesado la publicidad, la venta y la promoción de cosas. La psicología le fascinaba. Y estaba especialmente enamorado de la idea de elegir a las personas adecuadas para trabajar juntas de modo que su producción conjunta eclipsara la miserable suma de sus aportes individuales.

Dick estaba tratando de hacer eso en Informatics con su nuevo equipo de productos. Ya contaba con un auténtico mago en su equipo, un hombre delgado y tranquilo de unos cuarenta años llamado Jay Sullivan. Jay era un ex pianista de jazz que había llegado a Informatics procedente de un trabajo más mundano en su Chicago natal. Más tarde explicó por qué: “El software de sistemas [en Informatics] era mucho más interesante. No tenías que preocuparte por cosas mundanas como las solicitudes o las nóminas. Para mí era una programación mucho más real; te ocupabas más de la esencia de lo que era la programación. Las técnicas reales de programación son más importantes que los detalles del trabajo en un momento específico”. En otras palabras, podía hackear allí.

En su programación, Sullivan trabajaba como un veraneante que, tras haber planeado su viaje cuidadosamente, educándose sobre las características sutiles del paisaje local, seguía el itinerario con mayor conciencia. Sin embargo, todavía conservaba la curiosidad de desviarse del plan si las circunstancias parecían exigirlo, y obtenía placer de la exploración cuidadosa que implicaría semejante bifurcación en su camino, por no mencionar la sensación de logro cuando el desvío resultaba exitoso.

Al igual que muchos hackers, la inmersión de Sullivan en la programación tuvo sus consecuencias sociales. Sullivan explicó más tarde que con los ordenadores, “se puede crear un universo propio y hacer lo que se quiera en él. No hay que tratar con gente”. Así que, aunque era un maestro en su trabajo, Sullivan tenía la irritante personalidad de un programador que le hacía llevarse muy bien con los ordenadores, pero no prestaba demasiada atención a las sutilezas de la interacción humana. Insultaba a Dick con indiferencia y se dedicaba a sus asuntos con indiferencia, haciendo cosas brillantes con el sistema operativo, pero a menudo veía cómo sus innovaciones fracasaban porque no era experto en politiquería, un proceso necesario en la gran empresa. Dick Sunderland se había obligado a ser paciente con Sullivan y, finalmente, habían llegado a una relación de vendedor-inventor que produjo dos mejoras lucrativas para la línea Mark 4.

Dick buscaba más programadores expertos, llamaba a los reclutadores y dejaba muy claro que buscaba a gente de primera, nada menos. Un reclutador le mencionó a Ken Williams. “Este chico es un genio”, dijo el reclutador.

Sunderland llamó a Ken para una entrevista y se aseguró de que su verdadero genio, Jay Sullivan, estuviera allí para poner a prueba el temple de este tal Williams. Dick nunca antes había visto a nadie en igualdad de condiciones con Jay Sullivan, y tenía curiosidad por ver qué podía salir de la entrevista.

Dick y Jay estaban hablando de un problema en la implementación de un nuevo lenguaje fácil de usar en el que estaba trabajando Informatics cuando Ken apareció, vestido con pantalones y una camisa deportiva que le quedaban tan mal que era obvio que las camisetas eran su norma. La discusión había sido bastante técnica, centrándose en el problema de que para crear un lenguaje que un no programador pudiera entender (un lenguaje como el inglés) uno tendría que evitar cualquier tipo de palabras ambiguas o acrónimos.

De repente, Jay Sullivan se volvió hacia Ken y le dijo: "¿Qué piensas de la palabra 'any'?"

Sin dudarlo, Ken afirmó correctamente que era una palabra muy valiosa, pero ambigua de todos modos... y luego, improvisadamente, lanzó ideas sobre cómo se podría manejar esa palabra.

A Dick le pareció que estaba presenciando una batalla clásica: el descarado Pomona Kid contra el venerable Chicago Slim. Si bien Ken tenía una cualidad carismática y, obviamente, sabía de computadoras, Dick todavía tenía su dinero en Jay. Jay no lo decepcionó. Después de que Ken se detuvo, Jay, hablando tranquila y metódicamente, "cortó a Ken en pedazos con una hoja de afeitar", recordó Dick más tarde, enumerando los errores y la incompletitud de los pensamientos de Ken. Sin embargo, a Dick -y hasta a Jay- le impresionó que este desertor universitario pudiera siquiera tener esos pensamientos. Es más, en lugar de disuadirse por la andanada de Jay, Ken respondió de inmediato. Dick vio que los dos retomaban los hilos de las ideas del otro y los tejían en conceptos más refinados. Esto era sinergia, el santo grial del gerente, Dick decidió contratar a Ken Williams.

Dick puso a Ken bajo la supervisión de Jay, y los dos charlaban sobre arcanos de programación durante horas. Para Ken fue una educación: estaba aprendiendo la psicología de la informática de una manera que nunca había hecho. Por supuesto, una parte del trabajo que a Ken Williams no le gustaba era tener un jefe; Ken en este sentido era un hacker antiburocrático típico. Así que llegó a desagradarle Dick, con todos sus horarios y su fijación en los detalles gerenciales, obstáculos para el libre flujo de información.

Ken y Jay hablaban sobre las complejidades de algún aspecto del lenguaje de programación, como tratar de averiguar, cuando alguien dice "Listar por cliente", qué significa realmente. ¿Significa "ORDENAR por cliente", o tal vez "Listar TODOS los clientes"? ¿O tal vez "Listar CUALQUIER cliente"? (Esa palabra otra vez). El ordenador tenía que ser programado para que no se equivocara en ninguna de esas interpretaciones. Como mínimo, debería saber cuándo pedir a los usuarios que aclararan su significado. Esto requería un lenguaje de considerable flexibilidad y elegancia, y aunque Ken y su nuevo gurú Jay tal vez no lo hayan dicho en voz alta, una tarea de ese tipo va un poco más allá de la tecnología y entra en la lingüística primaria. Después de todo, una vez que uno se sumerge hasta el cuello en una discusión sobre el significado de la palabra “cualquiera”, sólo hay un pequeño paso para pensar filosóficamente sobre la existencia misma.

En algún momento en medio de una de estas conversaciones, Dick entraba, ansioso por presenciar alguna sinergia entre sus tropas. “Intentábamos hacer un supersubconjunto para que un niño de dos años lo entendiera, le pedíamos su opinión, él la daba y lo echábamos de la habitación”, recordó Ken más tarde. “Dick nunca entendió lo que estábamos planteando. Obviamente estaba fuera de su alcance”.

En esos momentos, Ken podría haberse sentido superior a Dick, pero en retrospectiva tuvo que admitir que Dick era lo suficientemente inteligente como para reconocer el talento. Ken se dio cuenta de que era uno de los miembros más débiles de un superequipo de programadores que estaban haciendo grandes cosas para Informatics. A veces, Ken pensaba que Dick debía haber tenido suerte, al reunir accidentalmente a cinco de las personas más creativas para su equipo de nuevos productos. O eso o era el mejor gerente del mundo, o al menos el mejor evaluador de talentos.

Ken, que siempre necesitaba más dinero, empezó a trabajar a tiempo parcial. Sunderland se negaba a sus constantes peticiones de aumentos de sueldo, y cuando Ken sugirió que le gustaría dirigir un grupo de programación, Dick, un poco sorprendido quizá por el descaro de este chico brillante pero descontrolado, rechazó de plano la petición. “No tienes talento para la gestión”, dijo Dick, y Ken

Williams nunca lo olvidó. Ken iba a casa de Roberta con regularidad y se quejaba de Dick (de lo malo que era, de lo estricto que era, de que no entendía a la gente y sus problemas), pero no era tanto una insatisfacción con su jefe como su deseo de más dinero, dinero para una casa más grande, un coche más rápido, una radio CB, una motocicleta, un jacuzzi, más aparatos electrónicos, lo que le llevó a duplicar e incluso triplicar el trabajo, a menudo pasando a un modo de no dormir. Con el tiempo, el trabajo externo llegó a ser más que el interno, y dejó Informatics en 1979, convirtiéndose en consultor independiente.

Primero había un tipo con un plan para hacer declaraciones de impuestos para grandes empresas como General Motors y Shell, y luego hubo un trabajo con Warner Brothers, programando un sistema para que la compañía discográfica mantuviera al día las regalías de los artistas. Había un sistema de contabilidad que construyó para Security Pacific Banks, algo sobre planes de impuestos extranjeros. Ken se estaba convirtiendo en un gurú de las finanzas; los treinta mil dólares al año que estaba ganando parecían ser solo el principio, si Ken seguía esforzándose.

Él y Roberta comenzaron a tejer una pequeña fantasía. Por la noche, las noches en que Ken no estaba fuera asesorando a alguien, se sentaban en el jacuzzi y hablaban de compartir la trampa suburbana de Simi Valley y mudarse al bosque. Donde irían a esquiar en el agua, esquiar en la nieve... simplemente a holgazanear. Por supuesto, no había tantas horas en un día para ganar dinero como para hacer ese tipo de trucos, sin importar para cuántas empresas Ken estableciera programas de impuestos. Así que la fantasía era solo eso, una fantasía.

Hasta que el hermano pequeño de Ken, Larry, recibió una computadora Apple.

Un día, Larry lo llevó a la oficina de Ken. A Ken, que había estado tratando con redes de telecomunicaciones que manejaban a dos mil personas a la vez, que había inventado lenguajes informáticos enteros con magos de mainframes como Jay Sullivan, la idea de que esa elegante máquina beige fuera un ordenador le parecía ridícula en cierto sentido. “Era un juguete comparado con los ordenadores que había estado usando”, explicó más tarde. “Un trozo de chatarra, una máquina primitiva”.

Por otro lado, había muchas cosas que ofrecía Apple que los gigantes descomunales de Ken no ofrecían. Hasta la época en que trabajaba en Informática, sus ordenadores se procesaban por lotes, cargando terribles tarjetas perforadas. Al menos, Apple era interactiva. Y, a fin de cuentas, era bastante potente, especialmente en comparación con las grandes máquinas de hace menos de una década. (Marvin Minsky, del MIT, calculó una vez que un Apple II tenía la potencia virtual del PDP-1.) Y funcionaba bastante rápido, casi comparable a una gran máquina, porque en un mainframe de tiempo compartido estás luchando por el tiempo de CPU con ochocientas personas intentando procesar su código a la vez, con la Bestia Estúpida sudando silicio intentando repartir nanosegundos entre cada usuario. No compartías tu Apple con nadie. En mitad de la noche, estaba ahí, en la casa, esperándote a ti y solo a ti. Ken Williams decidió que tenía que tener uno.

Así que en enero de 1980 juntó "cada centavo que tenía", como dijo más tarde, y compró un Apple II. Pero tardó un tiempo en comprender lo importante que era esa máquina. Ken se imaginó que todos los que tenían un Apple eran como él, técnicos o ingenieros. Parecía lógico que lo que realmente querían estas personas era un lenguaje potente para ejecutar en su computadora. Nadie había hecho todavía FORTRAN para el Apple. En ese momento, casi nadie había hecho nada con Apple, pero Ken pensaba como un hacker, incapaz de imaginar nada mejor que algo con lo que usar la computadora. El síndrome de las herramientas para hacer herramientas. (El primer gran proyecto

de Richard Greenblatt en el PDP-1 fue una implementación de FORTRAN, por la misma razón). En ese momento, Ken no podía concebir que Apple y pequeñas máquinas como esa hubieran abierto el campo de la informática recreativa a otros además de los hackers.

La ironía de esto era que, incluso cuando Ken planeaba escribir un FORTRAN para Apple, esta revolución más significativa en la informática estaba sucediendo allí mismo, en su propia casa.

• • • • • • •

Durante la mayor parte de su vida, Roberta Williams había sido tímida. Había en ella algo de ensueño, y sus ojos castaños de muñeca, su pelo largo y castaño y su vestuario femenino y con volantes (mangas acampanadas, botas de gamuza, cuellos Peter Pan) indicaban que se trataba de una mujer que había tenido una infancia llena de fantasía. De hecho, las primeras ensoñaciones de Roberta Williams habían adquirido proporciones casi sobrenaturales. Siempre se había imaginado a sí misma en situaciones extrañas. Por la noche, se acostaba en la cama y construía lo que ella llamaba "mis películas". Una noche, los piratas la secuestraban y ella ideaba elaborados planes de escape. En ocasiones era una historia en la que aparecía un salvador apuesto. Otra noche estaría en la antigua Grecia, soñando siempre con cosas que le sucedieran.

Hija de un inspector agrícola frugal del sur de California, era terriblemente tímida, y el relativo aislamiento de su hogar rural reforzaba esa actitud. “Nunca me agradé a mí misma”, reflexionaría más tarde. “Siempre quise ser otra persona”. Sentía que sus padres adoraban a su hermano menor, que sufría de epilepsia. Su forma de entretenimiento era contar historias que cautivaran a sus mayores y embelesaran a su hermano, que se las tomaba al pie de la letra. Pero a medida que se hizo mayor y se enfrentó a las citas y al mundo de los adultos, “todo eso se fue por la ventana”, como dice ahora. Cuando ella y Ken se casaron, esperaba pasivamente que él se ganara la vida; en cuanto a ella, era tan tímida que “apenas podía hacer una llamada telefónica”. La narración de historias quedó sepultada.

Una noche, Ken, que había traído una terminal de ordenador a casa, llamó a Roberta para mostrarle el programa que alguien había instalado en el ordenador central IBM al que estaba conectado. “Ven aquí, Roberta”, le instó, sentándose en el suelo alfombrado de color verde del dormitorio de invitados donde había puesto la terminal. “Mira esto, es un juego muy divertido”.

Roberta no quería saber nada de eso. En primer lugar, no le gustaban demasiado los juegos. En segundo lugar, era en un ordenador. Aunque Ken pasaba gran parte de su vida comunicándose con ordenadores, seguían siendo números poco amigables para Roberta. Pero Ken fue persistente y finalmente la convenció de que se sentara en el terminal para ver de qué se trataba. Esto es lo que vio:

> ESTÁS DE PIE AL FINAL DE UN CALLE, ANTE UN PEQUEÑO EDIFICIO DE LADRILLOS. A TU ALREDEDOR HAY UN BOSQUE. UN PEQUEÑO ARROYO FLUYE FUERA DEL EDIFICIO Y BAJA POR UN BARRANCO.

Se trataba de Adventure, el juego escrito en el laboratorio de inteligencia artificial de Stanford por el hacker Don Woods, el juego tolkieniano que atraía a hackers y usuarios a sumergirse en un mundo de mazmorras mágicas. Y desde el momento en que Roberta Williams probó GO EAST, quedó total e irrevocablemente enganchada. “No podía parar. Era compulsivo. Empecé a jugar y seguí jugando. Tenía un bebé en ese momento, Chris tenía ocho meses; lo ignoré por completo. No quería que me molestaran. No quería parar y hacer la cena”. No quería hacer nada más que

averiguar cómo llegar a Witt’s End o esquivar a la serpiente. Se quedaba despierta hasta las cuatro de la mañana, tratando de averiguar cómo esquivar a la maldita serpiente para llegar a las almejas gigantes. Y luego se sentaba en la cama pensando: ¿Qué no hice? ¿Qué más podría haber hecho? ¿Por qué no pude abrir esa estúpida almeja? ¿Qué hay dentro?

Al principio Ken participó, pero pronto perdió el interés. Roberta pensó que esto se debía a que a Ken nunca le había gustado que Adventure se pusiera sarcástico. Si decías MATAR AL DRAGÓN, te respondía: ¿QUÉ, CON TUS MANOS DESNUDAS? No podías enojarte, tenías que ignorarlo. Y ciertamente no podías responder con sarcasmo, solo decir: "Sí", y decía CON TUS MANOS DESNUDAS MATAS AL DRAGÓN Y ESTÁ MUERTO A TUS PIES. ¡Mataste al dragón! Podrías seguir. Roberta abordó el juego con una intensidad metódica, dibujando mapas elaborados y anticipando lo que había a cada paso. Ken pensó que era asombroso que un día Roberta no pudiera soportar las computadoras y al día siguiente no pudiera alejarla de la terminal. Finalmente, después de un mes de razonamiento sobre trolls, hachas, cavernas brumosas y vastos pasillos, Roberta resolvió Adventure. Estaba desesperada por encontrar más juegos como ese.

Para entonces, Ken había comprado una Apple. A pesar de su nuevo interés por los ordenadores, Roberta no estaba muy contenta con la compra de dos mil dólares. Si Ken tanto la quería, le dijo, debería intentar ganar dinero con ella. Esto coincidía perfectamente con los deseos de Ken en ese momento, que eran escribir un compilador FORTRAN para Apple y venderlo por fajos de dinero a los ingenieros y técnicos que querían herramientas para hacer herramientas. Contrató a cinco programadores a tiempo parcial para que le ayudaran a implementar el compilador. La casa de Ken, una típica casa de dos mil pies cuadrados de Simi Valley con cuatro dormitorios, se convirtió en la sede del proyecto FORTRAN.

Mientras tanto, Roberta había oído que había algunos juegos de estilo Aventura disponibles para Apple. Roberta compró algunos en una tienda de ordenadores en la cercana Northridge, en el Valle de San Fernando, pero le parecieron demasiado fáciles. Quería que su imaginación recién despertada fuera tan puesta a prueba y provocada como antes. Comenzó a esbozar un juego de aventuras propio.

Comenzó escribiendo una historia sobre una “casa misteriosa” y cosas que sucedían en ella. La historia tenía mucho que ver con *Los diez negritos* de Agatha Christie; otra inspiración fue el juego de mesa Clue. En lugar de simplemente encontrar tesoros como en Aventura, este juego te obligaba a hacer un trabajo de detective. Roberta trazó la historia tal como trazó un juego de aventuras cuando lo jugaba. A lo largo del camino, ideó rompecabezas, rasgos de personajes, eventos y puntos de referencia. Después de un par de semanas, tenía una pila de papeles con mapas y dilemas y giros y vueltas de tuerca en la trama, y ella lo dejó caer delante de Ken y dijo: “¡Mira lo que hice!”.

Ken le dijo a Roberta que su pequeño montón de papeles era muy bonito y que debería ir corriendo a terminarlo. Nadie quería realmente usar una computadora personal como una máquina de juegos; eran para ingenieros que querían descubrir cómo diseñar circuitos o resolver ecuaciones exponenciales de tres incógnitas.

Poco después, Ken y Roberta estaban en Plank House en el Valle, un restaurante de carnes con paredes de secuoya donde cenaban a menudo, y allí finalmente escuchó a su delicada esposa describir cómo su juego te colocaba en una vieja casa victoriana en la que tus amigos estaban siendo asesinados uno por uno. Ella describió algunos de los puzzles y habló de un pasadizo secreto. Comenzó a sonarle bien a Ken. Ken Williams generalmente podía oler algo de dinero por

ganar, y pensó que podría haber suficiente dinero en esto para un viaje a Tahití o para comprar muebles nuevos.

“Esto suena genial”, le dijo, “pero para vender de verdad se necesita algo más. Un ángulo. Algo diferente”.

Por casualidad, Roberta había estado pensando últimamente en lo genial que sería si un juego de aventuras estuviera acompañado de imágenes en la pantalla de la computadora. Se podría ver dónde se está en lugar de simplemente leerlo. No tenía idea de si esto era posible en una computadora Apple o en cualquier otro tipo. ¿Cómo se podría siquiera introducir una imagen en una computadora?

Ken supuso que podían intentarlo.

Por casualidad, acababa de salir al mercado un dispositivo llamado VersaWriter. Era una tableta en la que se dibujaba y registraba las formas en una computadora Apple. Pero no dibujaba con mucha precisión y era difícil controlar el mecanismo de escritura, que era como la base tosca de una lámpara de escritorio. Lo peor de todo es que costaba doscientos dólares. Ken y Roberta decidieron arriesgarse y comprarla. Luego Ken reprogramó todo el sistema para que Roberta pudiera hacer algo con él. Al final, hizo unas cuantas docenas de imágenes en blanco y negro de habitaciones dentro de la Casa Misteriosa, con personas dibujadas apenas mejor que monigotes. Luego Ken programó la lógica del juego, después de descubrir cómo comprimir setenta imágenes en un disquete, una tarea que cualquier programador mínimamente familiarizado con Apple habría supuesto imposible. El secreto no era almacenar datos de imágenes enteras, sino usar comandos en lenguaje ensamblador que almacenaban las coordenadas de las líneas individuales de cada imagen; a medida que aparecía cada nueva imagen, el ordenador seguía los comandos para dibujar la imagen. Era una asombrosa torpeza de programación que caracterizaba la facilidad de Ken para el hackeo de alto nivel.

Todo el proceso llevó un mes.

• • • • • • •

Ken descartó el proyecto FORTRAN y llevó el juego a una empresa distribuidora de software llamada Programma. Era el mayor distribuidor de software de Apple del mundo. A principios de 1980, eso no era decir demasiado. Vendía una gama de programas con nombres como Biorhythm, Nude Lady, Vegas Style Keno, State Capitals y Apple Flyswatter. La mayoría de los juegos estaban escritos en BASIC (en lugar del lenguaje ensamblador, que funcionaba mucho más rápido) y sólo podían entretener a un niño pequeño o a una persona enamorada de la idea de jugar con un ordenador. Había suficientes de estos últimos para aumentar la recaudación de Programma hasta 150.000 dólares al mes.

A la gente de Programma le encantó Mystery House. Se trataba de un juego de aventuras en lenguaje ensamblador que estaba bien planificado, era un reto y tenía imágenes. El hecho de que las imágenes fueran en blanco y negro y parecieran algo que podría haber dibujado el joven D.J. Williams (de seis años) era irrelevante. Nadie más había hecho nada parecido. Le ofrecieron a Ken un veinticinco por ciento de regalías sobre el precio de venta al por mayor de 12 dólares y le aseguraron que podían vender quinientas copias al mes durante seis meses, lo que a 3 dólares por copia serían 9.000 dólares. Esto era casi el doble de la cantidad que le habían prometido a Ken por

el compilador FORTRAN, antes de dividirla entre sus cinco programadores. Todo por el juego tonto de Roberta.

Ken Williams también consideró vender el juego directamente a Apple Computer. Envió una muestra, pero esperó más de un mes y no recibió respuesta. (Un año después, Apple - ahora una gran empresa con una burocracia que se movía lentamente - respondió y dijo que sí, que tal vez nos gustaría considerar comprarlo. Esto decía mucho más sobre en qué se había convertido Apple Computer que sobre Mystery House). Ken y Roberta no aceptaron la oferta de Programma. Ken y Roberta querían todo el dinero. ¿Por qué no intentar venderlo de forma independiente? Si no funciona, entonces recién entonces llevémoslo a Programma.

Así que los Williams comenzaron a llevar Mystery House a las pocas tiendas de computadoras de la zona. Al principio, la gente de las tiendas se mostraba escéptica: después de todo, los nuevos fanáticos de las computadoras, entusiasmados con el poder que les brindaban sus nuevas Apple, sus Radio Shack TRS-80 y sus computadoras PET, siempre estaban tratando de vender programas extraños. Pero luego, el juego de Roberta se iniciaba con una imagen de una casa antigua dibujada en la pantalla de alta resolución (high-res) de la computadora en lugar de la pantalla de baja resolución, torpe y orientada a bloques. Le preguntaban a Ken cómo lo había hecho. Después de algunas experiencias como esa, Ken y Roberta pensaron que podrían ganar hasta mil o dos mil dólares al mes con esta cosa de la venta de software.

El siguiente paso fue anunciar el producto en una revista. Pero mientras lo hacían, pensaron, ¿por qué no ofrecer un par de juegos más y parecer una empresa real? Ya tenían un nombre: On-Line Systems, un remanente de la visión de Ken de vender el tipo respetable de software empresarial para Apple que hacía en su trabajo de consultoría para empresas de computadoras en línea. Ken fue a ver a un amigo y le pidió que fuera el primer programador externo de On-Line. A cambio de las regalías que eventualmente recibiría, el amigo hizo un sencillo juego en blanco y negro de disparar un punto con otro punto llamado Skeet Shoot. Imprimieron algunos folletos publicitarios y hojas de documentación; no queriendo pagar la tarifa de cien dólares por la composición tipográfica, Roberta recortó las letras individuales de revistas y mandó imprimir ese "master" en una copistería local. El disco volvió con unas líneas que delataban su origen, copiado y pegado, pero ya habían gastado quinientos dólares. De todos modos, esa forma de empaquetado era lo último en tecnología en ese momento. Era el mundo de la informática, donde el empaquetado no importaba. Lo que importaba era la magia que se producía cuando se hacían todas esas conexiones binarias. El marketing era secundario a la sustancia.

Mystery House, o “Hi-Res Adventure #1”, tenía un precio de 24,95 dólares. Ken y Roberta, en un arranque de optimismo, habían comprado una caja de cien discos vírgenes en la tienda Rainbow Computing cercana, y una vez que enviaron los folletos a las tiendas de informática y colocaron el anuncio por una tarifa de doscientos dólares pagada a regañadientes en el número de mayo de 1980 de una pequeña revista llamada MICRO, esperaron. El teléfono sonó ese primer día de mayo, y luego hubo una pausa y luego volvió a sonar. Y a partir de entonces, pasaría mucho tiempo antes de que Ken y Roberta pudieran contar con que su teléfono no sonara.

Ken y Roberta ganaron once mil dólares aquel mes de mayo. En junio, veinte mil dólares. En julio, treinta mil. Su casa de Simi Valley se estaba convirtiendo en una máquina de hacer dinero. Ken se iba a trabajar a Financial Decisions, donde ahora programaba por unos cuarenta y dos mil dólares al año, y Roberta copiaba discos y los metía, junto con los volantes y los folletos, en una bolsa Ziploc.

También cuidaba de los niños, metía los programas en cajas, mantenía la casa limpia y despachaba los programas por UPS. Por la noche, Roberta diseñaba un juego de aventuras más largo y mejor basado en el mundo de los cuentos de hadas.

Cada pocos minutos sonaba el teléfono y lo más probable era que fuera alguien dispuesto a morir por completo a menos que recibiera una pista que lo sacara de una situación aparentemente desesperada en Mystery House. Las personas que llamaban al número que aparecía en el folleto incluido en la bolsa Ziploc con el disquete tenían la impresión de que On-Line era un gran conglomerado, y no podían creer la suerte que tenían de conectar de alguna manera con el verdadero autor del programa. —¿Estoy hablando con la persona que escribió el juego? —Sí, en su cocina. Roberta les daba una pista —nunca una respuesta directa: parte de la diversión era resolverlo por uno mismo— y charlaba con ellos un rato. El nivel de energía era contagioso. La gente se estaba volviendo loca por jugar con computadoras.

Ken Williams tenía una carga de trabajo completa en Decisiones Financieras, desarrollando un sistema financiero complicado y dirigiendo el departamento de procesamiento de datos. Por la noche, trabajaba en la Apple, hackeando un nuevo sistema de lenguaje de máquina para el nuevo juego de aventuras de Roberta. Los fines de semana, Ken hacía la ronda de las tiendas de computadoras. Estaba claro que el negocio del software requería todo su tiempo.

Roberta pensó que, ya que Ken estaba pensando en dejarlo, bien podrían vivir su viejo sueño de mudarse al bosque. Sus padres vivían cerca de Yosemite, sobre el pueblo de Oakhurst, y era incluso más rural y tranquilo que el lugar en el que Roberta creció y todavía recordaba con cariño. Sería perfecto para los niños. Así que lo hicieron. “Me voy a mudar a las montañas”, le dijo a un asombrado Dick Sunderland en una fiesta a mediados de 1980. Dick y Ken estaban en una habitación un poco alejada del ruido de la fiesta, y Ken dijo: “Aquí estoy, con veinticinco años, y la computadora Apple me ha permitido cumplir mi sueño: vivir en el bosque, en una cabaña de troncos y escribir software”.

Ken y Roberta compraron la primera casa de campo que vieron, una cabaña rústica de madera de tres habitaciones en Mudge Ranch Road, en las afueras de Coarsegold, California.

Para entonces, ya habían terminado el juego de cuentos de hadas de Roberta, El mago y la princesa. Era el doble de largo que La casa misteriosa, y funcionaba más rápido gracias a las mejoras de Ken en la lógica del programa. Ken había desarrollado un nuevo intérprete de lenguaje ensamblador para escribir juegos de aventuras; lo llamó ADL, o Lenguaje de Desarrollo de Aventuras. Además, esta “Aventura de alta resolución n.° 2” tenía más de ciento cincuenta imágenes. Ken había ideado subrutinas que permitían a Roberta introducir las imágenes en el programa con la misma facilidad con la que se dibujaba en una tableta normal. Esta vez, las imágenes eran en color; Ken utilizó una técnica llamada de sombreado llamada "dithering" para entremezclar los seis colores de Apple, en tramados de puntos, hasta obtener veintiún colores. Estaba realizando acrobacias en la Apple que Steve Wozniak jamás había soñado. Cosas mágicas.

El único problema del juego era el primer rompecabezas, en el que el aventurero, en su camino para rescatar a la princesa Priscilla de Serenia del mago Harlin, tenía que superar una serpiente. La respuesta era bastante confusa: tenías que coger una piedra y usarla para matar a la serpiente, pero a menos que eligieras una piedra en un lugar específico (todas se parecían) te mordía un escorpión y morías. La mayoría de la gente empezaba a darse cabezazos contra la pared a la tercera o cuarta picadura de escorpión. Finalmente, después de que innumerables aventureros frustrados llamaran a

la cocina de Roberta en Coarsegold (la gente de la Costa Este a veces llamaba a las 6 a.m., hora de California), On-Line comenzó a proporcionar una pista sobre ese dilema en cada paquete.

Con o sin serpiente, El mago y la princesa finalmente vendió más de sesenta mil copias a $32.95. Ken y Roberta se sentaban en el jacuzzi que habían instalado y movían la cabeza, diciendo: "¿Puedes creerlo?".

El 1 de diciembre de ese primer año, después de que el negocio ya les había cambiado la vida, les había conseguido una casa nueva y los había convertido en las estrellas en ascenso del mundo Apple, finalmente trasladaron el negocio fuera de la casa a un espacio en el segundo piso de un edificio en Oakhurst, siete millas por la Ruta 41. Su vecino era un promotor religioso que estaba tratando sin éxito de contratar a Little Richard para una gira nacional de predicación. Se le podía oír gritar a través de las delgadas paredes.

A principios de 1981, menos de un año después de que la empresa comenzara con unos cuantos disquetes y un anuncio de 150 dólares en una pequeña revista, Roberta describió la situación en una carta a otra pequeña revista: “Abrimos una oficina el 1 de diciembre de 1980 y contratamos a nuestra primera empleada para que nos ayudara con los despachos y los teléfonos. Dos semanas después, contratamos a alguien mas para que la ayudara, una semana después de eso contratamos a alguien para que los ayudara. Acabamos de contratar a un programador a tiempo completo esta semana y necesitamos al menos otro programador. Nuestro negocio está creciendo a pasos agigantados y no se vislumbra un final”.

Capítulo 15. La Hermandad

La ética hacker estaba cambiando, al mismo tiempo que se extendía por todo el país. Sus emisarios eran los ordenadores pequeños y de bajo coste que vendían Apple, Radio Shack, Commodore (el PET) y Atari. Cada uno de ellos era un ordenador de verdad; la mera proliferación creó una demanda de programas más innovadores que los métodos de distribución anteriores no podían satisfacer. Un hacker ya no podía distribuir programas ingeniosos dejándolos en un cajón, como hacía en el MIT, ni podía confiar en un sistema de Homebrew Computer Club de intercambio de programas en las reuniones del club. Muchas personas que compraron estos nuevos ordenadores nunca se molestaron en unirse a los clubes. En cambio, dependían de las tiendas de ordenadores, donde pagaban con gusto por los programas. Cuando estabas desesperado por algo que cumpliera la promesa de esta nueva y emocionante máquina, gastar veinticinco dólares en Mystery House parecía casi un privilegio. Los propietarios pioneros de ordenadores de principios de los años ochenta tal vez aprendieran lo suficiente sobre sus máquinas para apreciar la belleza de un flujo de información sin trabas, pero la ética hacker, al estilo de las microcomputadoras, ya no implicaba necesariamente que la información fuera libre.

A medida que empresas como On-Line escribían y vendían más programas, las personas que no tenían ningún deseo de convertirse en programadores, y mucho menos en hackers, comenzaron a comprar ordenadores, con la única intención de ejecutar software empaquetado en ellos. En cierto modo, esto representó la realización del sueño hacker: cómputo para las masas, ordenadores como tocadiscos: uno iba a la tienda de software, elegía los últimos lanzamientos y los ponía a girar. Pero, ¿realmente se beneficiaba de su ordenador si no lo programaba?

Aun así, a principios de los años ochenta, todo el que tenía un ordenador tenía que adentrarse en la mentalidad hacker hasta cierto punto. Hacer las cosas más simples en su máquina requería un proceso de aprendizaje, una búsqueda de gurús que pudieran decirle cómo copiar un disco o encontrar los cables de conexión adecuados para conectar la impresora. Incluso el proceso de compra de software listo para usar tenía un aire extraño y hacker. Los programas venían empaquetados en bolsas Ziploc, los gráficos de la llamada documentación estaban en su mayoría al nivel de los primitivos monigotes de Roberta Williams y, la mayoría de las veces, las etiquetas del disco estaban escritas a máquina y pegadas a mano... había un aura de ilegitimidad en el producto, sólo ligeramente más respetable que los libros de pornografía dura.

Una excursión a la tienda de informática local era un viaje a lo desconocido. El vendedor, que la mayoría de las veces era un joven que trabajaba por el salario mínimo, te tomaba la medida, como si fueras un obstáculo potencial en un juego de aventuras, y te ponía a prueba soltando la jerga de Ks, bytes, nibbles y tarjetas RAM. Intentabas que te explicara, por ejemplo, por qué este paquete de contabilidad funcionaba mejor que aquel, y él te respondía con algún galimatías sobre protocolos y macros. Finalmente, le harías la pregunta que casi todos los propietarios de Apple hacían en 1980 o 1981: “¿Cuál es el nuevo juego de moda?” Los juegos eran los programas que más aprovechaban la potencia de la máquina: ponían al usuario al mando de la máquina, lo convertían en el dios de los bits y bytes dentro de la caja (aunque no estuviera seguro de la diferencia entre un bit y un byte). El chico suspiraría, asentiría, buscaría bajo el mostrador el fenómeno de las bolsas Ziploc y, si tenías suerte, lo iniciaría en la pantalla y jugaría unas cuantas rondas, para poder ver lo que estabas comprando. Luego, desembolsarías tus veinte, veinticinco o incluso treinta y cinco dólares y te irías a casa a buscar lo que era la interfaz esencial con Apple: jugar jueguitos.

A principios de 1980, el nuevo juego de moda probablemente estaría escrito en un BASIC mortíferamente lento. La mayoría de las Apple de esa época usaban grabadoras de casetes; La dificultad de utilizar un ensamblador con una grabadora de casetes hacía casi imposible llegar hasta los más profundos confines de la máquina, el chip 6502, para hablar en el lenguaje ensamblador de Apple.

Esto estaba cambiando: Steve Wozniak había hackerado recientemente un diseño brillante para una interfaz de unidad de disco para Apple, y la empresa podía ofrecer unidades de disquete de bajo coste que accedían a miles de bytes por segundo, lo que facilitaba el ensamblaje a los pocos que sabían programar a ese nivel tan difícil. Por supuesto, los infectados con el imperativo práctico pronto se unirían a esa élite para aprender el sistema en su nivel más primal. Los programadores, los aspirantes a programadores e incluso los usuarios que compraban Apple invariablemente compraban unidades de disco junto con ellas. Como la Apple de Steve Wozniak se adhirió a la ética hacker en el sentido de que era una máquina totalmente “abierta” - con una guía de referencia de fácil acceso que te decía dónde estaba cada cosa en el chip y la placa madre - Apple era una invitación abierta a arremangarse y llegar al código hexadecimal del nivel de máquina. A hackear.

Así que Ken Williams no fue el único que se subió al tren de la gloria al hackear el lenguaje de máquina de Apple en la primavera de 1980. Los pioneros tecnológicos de todo el país estaban sintiendo lo que los hackers habían sabido desde siempre: las computadoras podían cambiar tu vida. En Saramento, un veterano de Vietnam llamado Jerry Jewell, que tenía el pelo rubio rojizo, un bigote a juego y una mirada perpetuamente aturdida y ligeramente enojada, había comprado una Apple para ver si podía saltar del negocio de los seguros a algo más lucrativo. Dos semanas después de conseguir la máquina, se inscribió en una clase de lenguaje ensamblador en el Lawrence Hall of Science impartida por Andy Herzfeld, uno de los mejores programadores de Apple. Jewell no tenía unidad de disco y no podía ejecutar los programas de muestra que se distribuían cada semana. Durante ocho semanas, no tuvo la menor idea de lo que Herzfeld estaba hablando, y ni siquiera breves tutoriales del instructor asistente - John Draper, alias Captain Crunch - pudo descifrar el código. Finalmente, después de que Jewell consiguiera una unidad de disco y escuchara las cintas que había grabado de la clase, lo entendió.

Jewell consiguió un trabajo como gerente de una tienda de computadoras local. En aquellos días, a las tiendas de computadoras acudía todo tipo de personas. Era casi como una declaración condicional en BASIC: SI tienes una computadora, ENTONCES probablemente estés un poco loco. Porque incluso entonces - transcurridos cuatro años de la Altair - todavía no se podían realizar muchas tareas útiles con un ordenador personal. Había un sencillo programa de procesamiento de textos llamado "Easy Writer", escrito por John Draper (Jewell compró una de las primeras copias en la Feria de Informática de 1980), y algunas cosas de contabilidad.

Pero la mayoría de la gente hackeaba Herramientas para Hacer Herramientas. O juegos. Y acudían a las tiendas de informática para mostrar sus hackeos.

Así que no fue sorprendente que un estudiante universitario de aspecto árabe llamado Nasir Gebelli se acercara a Jewell en la tienda y abriera un programa de presentación de diapositivas que había escrito. A Jewell le gustó y trabajó con Gebelli para crear un programa derivado, un programa de dibujo gráfico al que llamaron "E-Z-Draw". Jewell empezó a recorrer las tiendas de informática de Los Ángeles y el Área de la Bahía para venderlo.

Entonces, Nasir - un estudiante de informática que no estaba obteniendo buenos resultados en sus clases - empezó a escribir juegos. El uso que Nasir hacía del color y de una técnica llamada "Page Flipping" hacía que la actual generación de juegos tuviera un aspecto enfermizo. Page Flipping utilizaba una pantalla duplicada ("página") para todo lo que se mostraba en la Apple; recurriendo a instrucciones en lenguaje de máquina, se pasaba de una página a otra miles de veces por segundo, para eliminar el parpadeo que hacía que los gráficos de microcomputadoras parecieran tan poco atractivos. Nasir tampoco tenía miedo de utilizar cualquier cosa como "invasor" en sus juegos, que casi siempre utilizaban un escenario básico: había que disparar a muchas cosas antes de que algunas de ellas te dispararan de vuelta. Recreaba el estado de sitio adictivo y pirotécnico que era enormemente popular en los juegos arcade que funcionaban con monedas, que tenían microchips especiales para crear efectos gráficos espectaculares, y sólo cuando Nasir los mostró la gente se dio cuenta de que algunos de estos efectos se podían lograr en la Apple.

Ese año Nasir escribió doce juegos. Jewell y el dueño de la tienda de computadoras formaron una empresa llamada Sirius Software para venderlos. Jewell miraba la versión preliminar de Nasir de un juego y sugería cambios extravagantes. Un juego que Nasir escribió era bastante similar a Space Invaders - un popular arcade que funcionaba con monedas en el que los extraterrestres avanzaban por la pantalla en oleadas para atacar el pequeño tanque del jugador. Jewell sugirió que las armas disparadas por los invasores no deberían ser caparazones, sino huevos, y los invasores deberían ser, a su vez, monstruos, lobos espaciales, labios gigantes que lanzaran bombas y, lo más peligroso de todo, bolas de pelusa asesinas. Bolas de pelusa asesinas que rebotan, se sacuden y se mueven hacia ti con frenética inevitabilidad. *Space Eggs* fue un éxito de ventas desbocado para Sirius Software.

• • • • • • •

Otra empresa que irrumpió en el mercado en aquel entonces fue la creación de un ex abogado corporativo de Wisconsin. Doug Carlston no se sentía feliz trabajando en un gran bufete de abogados en el piso ochenta y dos del edificio Sears de Chicago; echaba de menos sus días de hacker en la universidad - cuando él y sus amigos metían chicles en la cerradura de la puerta de la sala de ordenadores para que el personal no pudiera impedirles entrar. Por la noche, quince de sus amigos entraban a escondidas y hackeaban. Incluso después de haber montado un pequeño bufete de abogados en la zona rural de Maine, su corazón seguía estando en la informática. Entonces, Carlston - de voz suave y contemplativo - se enteró de que Radio Shack vendía un ordenador por menos de dos mil dólares. Compró uno un viernes y no salió a tomar aire – recuerda - hasta ese domingo por la noche. Con el tiempo, empezó a escribir un gigantesco juego de estrategia en el TRS-80, que involucraba todo un universo imaginario. Tu misión era proteger a los buenos interestelares: la Brøderbund (que en escandinavo significaba “hermandad”).

Era principios de 1980 y Carlston - como Williams y Jewell - veía su vida en el software. Contrató a su hermano Gary, que había estado trabajando en un puesto tan deseable que los hombres adultos se quedaban sin aliento cuando lo mencionaba: entrenador de un equipo de baloncesto femenino escandinavo. Juntos crearon Brøderbund Software para vender Galactic Saga. La idea era traducir el Saga de TRS-80 a Apple.

Al principio, el Saga no tuvo demasiado éxito. Los siete mil dólares con los que empezaron Doug y Gary se redujeron a unos treinta y dos dólares en un momento dado. Vivían de la tarjeta VISA de Gary. No fue hasta que Doug atravesó el país en coche – deteniéndose en todas las tiendas de ordenadores que encontró y les mostrando el juego, permitiendo empaparse de algunos de los

puntos fuertes del programa - que consiguió diecisiete mil dólares de negocio, y pudo hacer una llamada nocturna a Gary avisando que las cosas mejoraraban.

Pero el gran salto llegó en la Feria de Informática de 1980, donde los Carlston habían reunido el dinero para mostrar al Saga en un "microstand" de bajo coste, una innovación de Jim Warren para permitir a las pequeñas empresas - a menudo sin ánimo de lucro - exponer sin desembolsar los crecientes honorarios de los expositores en el piso principal. A un empresario japonés conservador le gustaron estos Carlston religiosos y de vida simple, y les permitió distribuir el trabajo de algunos programadores japoneses con los que trabajaba. Los juegos eran copias fieles de los actuales juegos de arcade que funcionaban con monedas. Y el primer programa de Apple que les dio fue una brillante imitación del juego de arcade *Galaxian* (lo llamaron descaradamente *Apple Galaxian*), que se convirtió en un enorme éxito, vendiendo decenas de miles de discos. Y aunque Brøderbund comenzó a reclutar programadores en los Estados Unidos para escribir juegos, durante meses la producción japonesa representó la mayor parte de su negocio.

On-Line, Brøderbund y Sirius fueron las empresas que más rápido crecieron de las docenas que surgieron para atender a los nuevos usuarios de computadoras, en particular a los de lo que llegó a conocerse como el Mundo Apple. La Programma, que antes era dominante, se había expandido demasiado y finalmente se fusionó con una empresa mayor, pero que no era una fuerza de mercado tan grande. Pero empresas más nuevas con nombres como Continental y Stoneware y Southwestern Data también salieron al mercado como potros salvajes. La característica distintiva de estas empresas era que - al igual que las empresas de hardware que se formaron a partir del Homebrew Computer Club - el impulso parecía ser tanto lanzar software al mercado como sacar provecho de una tendencia en ciernes. Salir al mercado parecía ser la mejor manera de mostrar los hacks propios.

Significativamente, una nueva revista que se identificó estrechamente con la nueva ola de empresas de software del mundo Apple fue fundada por personas que no tenían mucha experiencia en el mundo editorial, pero eran fanáticos proselitistas del ordenador Apple.

Margot Tommervik, una editora de libros de texto freelance de Los Ángeles con el pelo castaño largo y liso al más puro estilo refugiado de los años sesenta, había adorado los juegos mucho antes de tocar su primer ordenador. A principios de 1980, apareció en el programa de televisión Password y, a pesar de que la emparejaron con un par de personalidades de novelas televisivas que - como recordaría más tarde - "no tenían ni idea de que Virginia estaba al sur y New Hampshire al norte", salió de una "ronda relámpago" hábilmente jugada con quince mil dólares. Ella y su marido Al - un editor de Variety - hicieron una lista de cosas que hacer con el dinero y resultó que necesitaban el doble para hacer mella en la lista. Así que dijeron al diablo con todo y salieron a comprar un ordenador.

El ordenador doméstico más conocido en aquella época era el TRS-80. Pero mientras Margot y Al esperaban a un vendedor en el Radio Shack local, un empleado de la tienda, un chico que estaba de pie cerca de Al, le preguntó: “¿Qué es ese olor?”. Al era un hombre rechoncho, pelirrojo y de barba larga que parecía un cobrador de peaje en un puente de la Tierra Media, y era inimaginable imaginarlo sin su pipa de brezo. El chico, tal vez con una aversión al humo al estilo del MIT en su sangre de hacker, le dijo a Al Tommervik: “Señor, no debería fumar esa pipa, me está poniendo enfermo”. Los Tommervik salieron de Radio Shack y una semana después compraron una Apple.

Margot y Al, en sus propias palabras, “se volvieron adictos” a la Apple. Disfrutaba de los juegos que jugaba, pero su satisfacción era más profunda. Sin ningún conocimiento técnico, Margot

Tommervik fue capaz de extraer la ética hacker de esta elegante pieza de maquinaria que tenía en su casa. Ella creía que su Apple tenía su propia personalidad, amante de la vida y un poco loca, en un sentido positivo. Más tarde explicó: “La idea de llamarla Apple es maravillosa. Es mucho mejor que [darle un nombre como] 72497 o 9R. Dice: ‘Oye, esto es más que una simple pieza de maquinaria. Puedes sacarle más provecho’. Incluso el pequeño pitido que emite cuando la enciendes muestra un entusiasmo especial”.

Margot Tommervik conoció la historia de cómo comenzó Apple Computer y se maravilló de cómo la máquina transmitió el “espíritu amante de la vida” de Steve Wozniak al ordenador. Tenía esa capacidad de morder todos los grandes trozos de vida y masticarlos y saborear cada pedacito. Puso el espíritu en la computadora mientras la construía. Hizo que la máquina hiciera tantas cosas como se le ocurriera…”. Margot creía que si pasabas suficiente tiempo con tu Apple, te darías cuenta de que también podías hacer cualquier cosa que se te ocurriera. Para ella, Apple encarnaba la esencia del ser pionero, de hacer algo completamente nuevo, de tener el coraje y la voluntad de asumir riesgos, de hacer lo que no se había hecho antes, intentar lo imposible y lograrlo con alegría. La alegría de hacer que las cosas funcionen. En resumen, la alegría del hackerismo, por primera vez transparente para aquellos que no nacieron con el Imperativo del Manos a la Obra.

Margot lo vio en todos los que usaban la Apple. Simplemente se enamoraron de ella. Su fontanero, por ejemplo, recibió una Apple, y mientras Margot observaba a la esposa del fontanero jugando un juego en ella, Margot juró que en realidad estaba viendo una mente en expansión. Uno podía sentir algo de esta emoción incluso con solo configurar una Apple, cuando conseguía arrancar su primer disco y la unidad de disco se encendía, zumbando alegremente, con la pequeña luz roja de "en uso" encendida. ¡Por Dios, lo habías logrado! Hiciste que algo sucediera. Hiciste que la unidad de disco funcionara, hiciste que esto sucediera, y luego, cuando comenzaste a establecer tareas reales para tu Apple y a construir tus pequeños universos, comenzaste a resolver cosas. Viste que tu poder aumentaba enormemente. Todas las personas con las que habló en el mundo de Apple, y ciertamente Margot misma, mostraron esa alegría. Ella creía que no era nada menos que la alegría de la propia humanidad.

A Margot Tommervik le encantaban los nuevos tipos de software que salían al mercado y, aunque ella y Al hacían algo de programación en BASIC, la máquina se utilizaba sobre todo para jugar a los nuevos juegos que ella compraba. Un día, pasó por Rainbow Computing y vio un anuncio de que iba a salir un nuevo programa de estilo aventura y que se pondría a la venta a las diez de la mañana de un viernes determinado; el primero que lo resolviera ganaría un premio. Margot estaba allí con 32,95 dólares ese viernes y al mediodía del sábado estaba de nuevo en la tienda con la solución. El juego era Mystery House.

Algún tiempo después, Margot se topó con una editorial que había empezado a publicar una revista sobre software y buscaba un socio. Margot y Al dijeron que pondrían algo de dinero y publicarían la revista si les prometían el control total. Así que el resto del dinero de *Password* se destinó a esta nueva encarnación de la revista, una revista dedicada al mundo de la computadora Apple. Se llamaría *Softalk*.

Cuando Margot empezó a buscar anunciantes, llamó a On-Line y le dijo a Roberta - que todavía se ocupaba de los asuntos corporativos desde su cocina de Simi Valley - que quería una revista completamente profesional que reflejara el espíritu de la computadora Apple. El entusiasmo de Margot era evidente. Y cuando Margot mencionó que había sido ella quien había ganado ese

concurso para resolver Mystery House, Roberta gritó: "¡Eres tú! Pensamos que llevaría meses hacerlo". Roberta habló con Ken y On-Line decidió sacar cuatro anuncios de un cuarto de página en el primer número. Llamaron a otras empresas y les instaron a que también sacaran anuncios.

Softalk salió en septiembre de 1980 con treinta y dos páginas, incluidas las portadas. Con el tiempo, la gente de la industria casera de suministro de productos para Apple comenzó a darse cuenta del valor de una revista cuyos lectores eran su público objetivo directo. A finales de 1981, había más de cien páginas de publicidad por número.

• • • • • • •

Estas empresas pioneras del mundo Apple estaban unidas por un vínculo espiritual tácito. Todas amaban la computadora Apple y la idea de la computación de masas en general. De alguna manera, todas creían que el mundo sería mejor cuando la gente tuviera en sus manos las computadoras, aprendiera las lecciones que las computadoras tenían que enseñar y, especialmente, consiguiera un software que ayudara a acelerar este proceso.

En pos de este objetivo común, On-Line, Sirius y Brøderbund se convirtieron casi en una Hermandad propia. Jewell, los Williams y los Carlston llegaron a conocerse muy bien, no sólo en las ferias de informática y eventos comerciales, sino también en las fiestas de cada uno, donde se reunían los tres equipos, junto con gente de otras empresas orientadas a Apple en California.

Esto contrastaba enormemente con algunas empresas no tan antiguas pero ya moribundas. En particular, Atari - la empresa que comenzó como el primer proveedor de juegos de ordenador y vendió millones de dólares en software para la máquina de juegos Atari "VCS" (que no se podía programar como un ordenador) - y su propio competidor de Apple, el Atari Home Computer. Desde su adquisición por el enorme conglomerado Warner Communications, Atari se había despojado de la apertura al estilo hacker de sus fundadores. Casi había que ser un agente de la KGB para averiguar el nombre de uno de sus programadores, tan aterrorizada estaba Atari de que alguien pudiera asaltar sus filas. Y la idea de que los programadores se reunieran y compararan notas era aún más aterradora. ¿Qué pasaría si uno de sus programadores se diera cuenta de que podía hacerlo mejor en otro lugar? No había secretos para la Hermandad, que en 1981 pagaba a sus programadores con una tasa de regalías del treinta por ciento, una tasa bien conocida por las tres empresas y todos los programadores que trabajaban en el campo.

La cooperación iba más allá de las fiestas. Casi como si inconscientemente se hubieran comprometido a adherirse al menos a una parte de la Ética Hacker, no había secretos entre ellos. Casi todos los días, Ken, Doug y Jerry hablaban por teléfono, compartiendo información sobre este distribuidor o aquel fabricante de disquetes. Si algún minorista no pagaba a una de las empresas, las demás lo sabrían inmediatamente y no se lo entregarían a ese minorista. “Teníamos este código no escrito”, recordó más tarde Jerry Jewell. “Nos comunicábamos mutuamente en qué estábamos trabajando para no hacer los mismos proyectos. Si yo estaba trabajando en un juego de carreras de coches, se lo decíamos, para que no empezaran uno”.

Algunos podrían considerar esta interacción como una restricción del comercio, pero esa sería una interpretación de la Edad Antigua. Esta Hermandad no era un cártel que se unía en detrimento del usuario y la tecnología. El usuario se beneficiaba al obtener una gama más amplia de juegos. Y si un programador de una de las empresas no podía descubrir algún truco en lenguaje ensamblador sin gráficos de página, el hecho de que pudiera ponerse en contacto con un programador de otra empresa era solo la aplicación de la Ética Hacker al comercio. ¿Por qué ocultar información útil? Si

los hacks ingeniosos se difundieran ampliamente, la calidad de todo el software aumentaría y la gente sacaría más provecho de las computadoras, lo que sería bueno para todas las empresas en el largo plazo.

Tal vez era hora de eliminar las prácticas divisivas de las empresas y adoptar un enfoque más parecido al de los hackers, que, gracias a sus éxitos en el campo del software, podría extenderse por todo Estados Unidos y revitalizar a todo el país, que desde hace tiempo gira en una vorágine darwiniana, litigiosa y dominada por los MBA. La sustancia podría entonces prevalecer sobre la turbia "imagen corporativa", en un mundo libre de la práctica insana y antiproductiva de poseer conceptos, donde los secretos comerciales podrían distribuirse por todas partes. Un mundo sin toda esa seriedad destructiva y despiadada. La actitud en el mundo de Apple parecía ser "si no es divertido, si no es creativo o nuevo, no vale la pena". Eso es lo que se oía decir a Ken y Roberta Williams, a Doug y Gary Carlston, a Jerry Jewell.

Este espíritu alcanzó su cenit durante el verano de 1981, en una escena que tenía todo el entusiasmo de un anuncio de gaseosas: un viaje en balsa por los rápidos del río Stanislaus. Fue idea de Ken Williams, un viaje de vacaciones conjunto para toda la industria. Ken bromeó diciendo que lo hacía sólo para hacer agujeros en las balsas de sus competidores; pero el propio absurdo de esa afirmación subrayaba la diferencia entre esta industria y otras. En lugar de sabotear a los competidores, Ken Williams se abriría paso a través de aguas turbulentas junto a ellos.

El río era idílico, pero un participante explicó más tarde a un periodista que aún más idílico que el entorno aislado entre pinos y altas paredes de cañón era el sentimiento entre los aventureros, que por supuesto intercambiaban todo tipo de información sobre productos, tecnología y finanzas: "Todos sentimos que hemos vencido al sistema: llegamos a los microordenadores antes que IBM. Todos somos competidores, pero nos gusta cooperar".

Incluso los barqueros tuvieron que pedir a los participantes - entre los que se encontraban los jefes de más de seis empresas de software, como Ken y Roberta, o los Carlston, o Steve Dompier (el miembro de Homebrew que estaba escribiendo software de forma independiente ahora que Processor Tech había cerrado el negocio) - que dejaran de hablar de trabajo. A veces se detenían. Se detenían al final del recorrido cuando se acercaban al último rápido. No era la primera vez que Ken Williams embistió con su balsa a otra. Algunas personas de esa balsa cayeron sobre otra, y la gente de las diez balsas usó sus remos y baldes para salpicarse unos a otros, y la Hermandad explotó en una niebla de agua blanca, risas y una camaradería emocionante.

Capítulo 16. La tercera generación

Todavía existían los hackers natos, aquellos bendecidos con una curiosidad incansable, el Imperativo de la Manos a la Obra. Los últimos elegidos en baloncesto y los primeros en la clase de aritmética para adivinar los misterios de las fracciones. Los alumnos de quinto grado que murmuraban, cuando los adultos les presionaban para que se explicaran, que les "gustaban los números". Los niños con mechones de pelo en el fondo del aula de secundaria que se adelantaban tanto a la clase que los profesores de matemáticas se daban por vencidos con ellos, les dejaban saltar a capítulos posteriores del texto y finalmente les permitían salir del aula y bajar las escaleras para descubrir, con el mismo asombro con el que Peter Samson se topó con la sala EAM del MIT, una terminal conectada a un ordenador de tiempo compartido en alguna universidad. Una terminal de teletipo gris en el sótano de una escuela suburbana, una terminal que contenía, maravilla de maravillas, juegos. Podías jugar a los juegos, pero si habías nacido hacker, eso no sería suficiente. Te preguntarías: "¿Por qué el juego no puede hacer esto?" “¿Por qué no puede tener esa característica?” Y como se trataba de una computadora, por primera vez en tu vida tendrías el poder de convertir esto en aquello. Alguien te enseñaría algo de BASIC y el sistema estaría a tus órdenes.

Pasó exactamente lo mismo con John Harris. Aunque era alto y atractivo, un rubio de pelo claro con una sonrisa ridículamente atractiva y la forma de hablar entrecortada de alguien cuyo entusiasmo es demasiado alto para reconocer interrupciones gramaticales que hacen perder el tiempo, era un paria social. Más tarde admitiría alegremente que había sido “el peor estudiante de inglés de la escuela y el peor en Educación Física”. Sus raíces estaban en la clase media alta de San Diego. Su padre era funcionario bancario. Sus hermanos, un hermano menor y dos hermanas gemelas mayores, no estaban interesados en cuestiones técnicas. “Yo era completamente técnico, cien por ciento”, dijo John más tarde con una redundancia entrañable. Parecía que no tenía un confidente más íntimo que la computadora remota (ni siquiera sabía su ubicación) conectada a la terminal de tiempo compartido de su escuela.

John Harris no era uno de esos genios metódicos y lentos que deslumbraban a la gente en las ferias de ciencias. Impresionar a los adultos no era su fuerte. El arte de John Harris dependía de impresionar a la gente que compartía sus pasiones, que eran pocas y bien definidas: ciencia ficción (películas y cómics, no libros, porque John no era un gran lector), juegos y hacking.

En algún momento, la cumbre de la existencia para una persona como John Harris podría haber sido encontrar el camino hacia un centro de computación como el laboratorio de inteligencia artificial del MIT, donde habría holgazaneado y aprendido hasta que tuvo la oportunidad de usar una terminal. Podría haber sido como un envío al cielo, como lo había sentido David Silver, de catorce años, cuando fue iniciado por los hackers del noveno piso y se le permitió tomar el sacramento del PDP-6. Pero Harris llegó a la edad de la escuela secundaria después de la revolución que comenzó con el Altair. La generación de John Harris fue la primera que no tuvo que mendigar, pedir prestado o robar tiempo de computadora de una computadora central distante conectada a terminales de teletipo. En los suburbios exuberantes alrededor de San Diego, no era raro que un estudiante de secundaria en 1980 engatusara a sus padres, o incluso ganara suficiente dinero con un trabajo a tiempo parcial, para una gran compra. La mayoría de los niños querían autos. Pero como bien sabían los primeros dueños de tiendas de computadoras, otros niños pedían computadoras.

Cuando John Harris estaba en undécimo grado, un estudiante de último año que conocía le dejó usar su computadora Commodore PET. John recordó más tarde: "Empecé a jugar juegos en su sistema y

comencé a programar en su sistema, un juego de Star Trek. Y un par de cosas más en BASIC que había aprendido y que eran mucho más divertidas que cualquiera de las cosas de tiempo compartido. Era más rápido, era mucho más interactivo, tenía gráficos y efectos de sonido... Los teletipos estaban bien, pero no sabía que existía nada más, y pensé: 'Vaya, esto es genial...' .'"

Para la Tercera Generación de John Harris, que siguió a la generación pionera de hackers de mainframes y a la segunda generación de hackers de hardware que liberaron los ordenadores de las instituciones, el acceso a los ordenadores era fácil. Podías tener uno o usar el de un amigo. Los ordenadores no eran tan potentes como los de las instituciones y no había comunidades de magos, ni Greenblatts ni Gospers que te instaran a abandonar la perdición y dedicarte a lo correcto hasta que pudieras ser considerado un ganador. Pero esos hechos de la vida no molestaban a esta Tercera Generación. Ahora podían tener ordenadores en sus habitaciones. Y todo lo que aprendieran sobre hacking, y cualquier elemento de la Ética Hacker que adquirieran, estaría determinado por un proceso de aprendizaje que se desarrollaría a partir del hacking en sí.

John Harris estaba fascinado con el PET. Podías hacer las cosas mucho más fácilmente con un ordenador personal. John estaba particularmente impresionado con la capacidad de edición a pantalla completa, una gran mejora con respecto al proceso de edición estilo teletipo de una línea a la vez en el que se había estancado antes. Pero lo mejor del PET y de otros ordenadores personales eran los juegos.

"Estoy obsesionado con todo tipo de juegos", dijo más tarde John Harris. "¡Supongo que así soy yo!". Era natural que un adicto a la electrónica de secundaria se sintiera deslumbrado por la tanda de juegos arcade de guerra espacial que aparecieron a finales de los setenta: Harris no sabía que su inspiración era el hack Spacewar de Slug Russell. Durante un tiempo después de eso, John se enamoró de un juego llamado Crazy Climber, en el que intentas llevar a un tipo a lo alto de un edificio, evitando macetas que se caen, personas que cierran ventanas en tu mano y un gorila gigante que intenta aplastarte. Lo que le impresionó de Crazy Climber fue su innovadora creación de un escenario único y artístico. Hizo algo que nadie había hecho antes.

John Harris se esforzó por alcanzar ese nivel de originalidad. Su actitud hacia los juegos era similar a su actitud hacia los lenguajes informáticos o su preferencia por un determinado ordenador sobre otro: una intensa identificación personal y una tendencia a ofenderse por una forma ineficiente y subóptima de hacer las cosas. John llegó a la conclusión de que los juegos debían tener un cierto grado de innovación, un cierto grado de deslumbramiento gráfico y un cierto grado de desafío. Sus estándares de "jugabilidad" eran rígidos. Se ofendía personalmente en los casos en los que un programador podría haber mejorado el juego de alguna manera obvia (para John Harris), pero no lo hizo, ya sea por ignorancia técnica, un lapsus de percepción o, lo peor de todo, por pereza. Los detalles hacían que un juego fuera realmente genial, y John adoptó la firme creencia de que un autor de juegos debía incluir todos los detalles posibles para que el juego fuera más divertido. Sin descuidar, por supuesto, el perfeccionamiento de la estructura básica del juego para que estuviera esencialmente libre de errores.

Para cumplir con sus propios estándares exigentes, John necesitaba su propio ordenador. Empezó a ahorrar dinero. Incluso redujo el tiempo que pasaba jugando a los juegos de arcade. Para entonces, John ya había terminado la escuela secundaria, se había matriculado en una universidad local de ingeniería eléctrica y estaba trabajando en el centro de procesamiento de datos de un banco. Uno de

sus amigos tenía el ordenador doméstico más moderno para hackers, el Apple, pero a John no le gustaban las capacidades de edición de la máquina ni sus gráficos extravagantes.

Con dinero en la mano fue a comprar un ordenador, en busca de un PET. Los vendedores se burlaban de él. “La única persona que compra un PET es una persona que está hasta el último centavo”, le decían. “Una persona que no puede permitirse un Apple II”. Pero John Harris no quería la creación de Wozniak. Había visto más del Apple de su amigo y estaba más convencido que nunca de que el Apple tenía graves daños cerebrales. Su desprecio por el Apple creció más allá de todo límite. “Incluso ver ese ordenador me pone de los nervios”, diría más tarde. Ante la mera mención de la máquina, Harris retrocedía y hacía la señal de la cruz, como si estuviera alejando a un vampiro. Podía explicar con detalle por qué se sentía así: no tenía editor de pantalla completa, la necesidad de cargar la máquina con más hardware antes de que funcionara de verdad, el teclado limitado... Pero este odio iba más allá de la razón. De alguna manera Harris sentía que Apple le impedía hacer lo que quería hacer. Mientras que otros hackers consideraban que las limitaciones de Apple eran obstáculos difíciles de superar o un susurro seductor que decía: “Llévame más lejos”, Harris las consideró ridículas. Así que le preguntó al vendedor de una de las tiendas sobre esta otra máquina, el ordenador Atari.

Atari acababa de sacar al mercado su 800 (y su compañero de menor potencia, el 400), su competidor de Apple. A primera vista, parecía una especie de consola de juegos con un teclado. De hecho, tenía una ranura para poner cartuchos en su interior, una señal de que la máquina estaba orientada, al menos en parte, a novatos demasiado confundidos para manejar incluso un casete de cinta, y mucho menos un disquete. Ni siquiera había un manual decente. John Harris jugó con un 800 en la tienda y descubrió que, al igual que el PET y a diferencia del Apple, tenía edición a pantalla completa. Pero quería saber qué había dentro, así que fue a otra tienda, donde un vendedor le pasó un trozo de papel con algunos comandos para este nuevo ordenador. Como un código secreto para uso de la Resistencia francesa. Ningún descifrador de códigos devoraba un mensaje con tanta avidez como John Harris estos papeles. Descubrió que el Atari tenía un conjunto de símbolos gráficos de pulsación de teclas, un modo de alta resolución y un chip independiente para efectos de sonido. En resumen, nuevas y emocionantes funciones, todas las funciones que a Harris le gustaban del PET, e incluso las cosas que a regañadientes consideraba que valían la pena en el Apple. Compró un 800.

Empezó a programar en BASIC, pero muy pronto se dio cuenta de que tendría que aprender lenguaje ensamblador para hacer los juegos que quería hacer. Dejó de trabajar en el banco y consiguió un trabajo en una empresa llamada Gamma Scientific, que necesitaba un programador para realizar trabajos en lenguaje ensamblador en su sistema y estaba dispuesta a formar a alguien.

Transferir sus nuevas habilidades en lenguaje ensamblador al Atari fue difícil. El Atari era una máquina "cerrada". Esto significaba que Atari secuestraba la información relativa a los resultados específicos que se obtenían al utilizar el lenguaje ensamblador del microprocesador. Era como si Atari no quisieera que el Pueblo escribiese programas en ellas. Era la antítesis de la ética hacker. John escribía a la gente de Atari e incluso los llamaba por teléfono para hacerles preguntas; las voces en el teléfono eran frías, no ayudaban. John pensó que Atari estaba actuando de esa manera para suprimir cualquier competencia con su propia división de software. Esta no era una buena razón para cerrar su máquina. (Diga lo que diga sobre Apple, la máquina era "abierta", sus secretos estaban disponibles para todos y cada uno). Así que John se quedó reflexionando sobre los misterios de Atari, preguntándose por qué los técnicos de Atari le dijeron que el 800 le daba solo cuatro

colores en el modo gráfico, mientras que en el software que lanzaron para él, juegos como Basketball y Super Breakout, había claramente más de ocho colores. Se decidió a descubrir sus secretos, los misterios de su sistema, para poder extenderlo y controlarlo mejor.

Para la búsqueda, John reclutó a un amigo que sabía lenguaje ensamblador. Consiguieron un desensamblador de cintas de casete escrito en BASIC, algo que descomponía los programas en su código objeto y desensamblaba el software vendido por Atari línea por línea. Luego tomaban estas extrañas instrucciones, que accedían a todo tipo de extrañas ubicaciones de memoria en el chip 6502 dentro del Atari, y las introducían en la máquina para ver qué pasaba. Descubrieron cosas como "interrupciones de lista de visualización", que permitían utilizar un mayor número de colores en la pantalla; "caracteres definibles por el usuario"; y, lo mejor de todo, algo que más tarde conocerían como "gráficos de misil de jugador", que no era nada menos que un método en lenguaje ensamblador para acceder a un chip especial de Atari llamado "Antic" que manejaba los gráficos por sí solo, lo que permitía ejecutar el resto del programa en el chip principal. Dado que uno de los aspectos más difíciles de programar juegos era dividir las actividades del chip principal entre sonido, gráficos y lógica del juego, los gráficos de misil de jugador le proporcionaban una enorme ventaja. ¿Cómo podía una empresa que había hecho algo tan ingenioso en su máquina ser tan tacaña al hacernos saber que existía?

Harris y su amigo habían descifrado los secretos de la Atari. Querían usar su conocimiento para liberar la máquina, distribuir los datos técnicos, abrir de par en par el mercado de Atari. Pero en esa época aparecieron unos manuales de hardware piratas. Parecía que algunos hackers dentro de Atari habían conseguido copias de su hardware interno y del manual de referencia y los estaban distribuyendo a precios elevados entre las partes interesadas. Sin embargo, el manual estaba escrito de tal manera que sólo las personas que ya eran el equivalente a los ingenieros de diseño de Atari podían adivinarlo. Como Harris dijo más tarde, "Estaba escrito en Atari, no en inglés". Así que el manual pirata no fue de mucha ayuda excepto para aquellas personas que habían integrado el funcionamiento de la Atari 800 en su propia cosmología mental. Personas como John Harris.

John Harris, de dieciocho años, utilizó este conocimiento para escribir juegos. Escribía juegos que le gustaría jugar, y su deseo de hacerlos lo suficientemente llamativos y emocionantes como para complacerlo como jugador lo incitó a aprender más sobre el sistema Atari. Como fanático de la ciencia ficción que asistía a menudo a las “Cons” (los cónclaves de fanáticos de la ciencia ficción, donde la gente perdida en la fantasía tecnológica era considerada normal), naturalmente gravitó hacia los juegos de guerra espacial. Creaba naves espaciales, estaciones espaciales, asteroides y otros fenómenos extraterrestres. A partir de su imaginación, hacía que estas formas aparecieran en su pantalla y luego las controlaba. Ponerlas en la pantalla y controlarlas era mucho más importante que el destino final del juego en sí: John Harris podía ser descuidado y a menudo perdía programas enteros al guardar archivos en el lado incorrecto de la cinta de casete o expandir el código para que el programa se bloqueara, y solo entonces se daba cuenta de que no había hecho una copia de seguridad en la cinta. Se sentía mal por ello, pero seguía hackeando.

Hackear era lo mejor en su vida. Había comenzado a trabajar a tiempo completo en Gamma Scientific para mantenerse. El sueldo era de menos de diez mil dólares al año. Le gustaba el trabajo en la medida en que le permitía trabajar con el ordenador. En casa tenía su 800, ahora equipado con una unidad de disco para una programación sofisticada en lenguaje ensamblador. Pero sin una comunidad tan unida como la que tenían los hackers del MIT, descubrió que hackear no era suficiente. Anhelaba más contacto social. Su relación con su familia era inestable. Más tarde afirmó

que lo "echaron" de su casa porque su padre tenía expectativas que John no podía igualar. Describe a su padre como poco entusiasta con su manía de programar juegos en un ordenador Atari 800. Así que Harris se mudó a una casa con unos cuantos fanáticos de la ciencia ficción. Asistía a las convenciones con ellos, eventos alocados en los que podían quedarse despiertos durante días seguidos, rondando los pasillos del hotel con pistolas de dardos de plástico. Pero a menudo a John le parecía que sus amigos estaban planeando alguna excursión interesante sin invitarlo. John Harris era un joven amistoso, de paso lento y muy sensible a estos aparentes rechazos.

Quería una novia. Las ocasiones aisladas en las que había salido con miembros de este género deseable pero elusivo siempre parecían terminar en algún tipo de decepción. Sus compañeros de casa a menudo se involucraban en intrigas románticas (llamaban en broma a la casa "Peyton Place of Outer Space"), pero John rara vez lo hacía. Había una chica con la que estuvo un par de semanas, e incluso había quedado para Nochevieja. Pero ella lo había llamado justo antes de Año Nuevo. "No sé cómo decirte esto", dijo, "pero conocí a un chico y me voy a casar con él". Eso era típico.

Así que siguió hackeando juegos. Al igual que los hackers del MIT o los Homebrewers, su recompensa era la satisfacción de hacerlo. Se unió a un grupo local de usuarios de Atari y tomó prestados programas de su biblioteca para que funcionaran más rápido y hicieran cosas interesantes. Tomó, por ejemplo, una versión del juego de arcade Missile Command y lo aceleró, amenizó las explosiones cuando uno de tus ICBM detuvo el arma nuclear enemiga que estaba destruyendo tu ciudad. Mostraba su trabajo a otros y ellos se divertían con él. Todo su hackeo automáticamente revertía al dominio público; la propiedad era un concepto con el que nunca se ocupó. Cuando alguien del grupo de usuarios le dijo a John Harris que tenía una pequeña empresa que vendía juegos de computadora y que le gustaría comercializar un juego de John, la reacción de Harris fue: "Claro, ¿por qué no?". Fue como regalar un juego y recibir dinero por ello.

Le dio al hombre un juego llamado Battle Warp, que se parecía mucho al antiguo MIT Spacewar, un juego para dos jugadores en el que las naves "vuelan y se disparan entre sí", como John lo describiría más tarde. Harris ganó alrededor de doscientos dólares con Battle Warp, pero fue suficiente para que pensara en distribuir su material más ampliamente que a través de la red de grupos de usuarios.

En marzo de 1981, Harris fue a la Feria de Computación de San Francisco, principalmente para asistir a un seminario sobre programación de Atari impartido por uno de los mejores programadores de Atari, Chris Crawford. John quedó muy impresionado con Crawford, un tipo tímido que se movía de un lado a otro cuando hablaba y era hábil para explicar las cosas. Después de eso, John Harris estaba entusiasmado, deambulando por los pasillos densamente abarrotados de Brooks Hall, mirando todas las máquinas nuevas y echando un vistazo a las docenas de nuevas empresas de software que habían ocupado puestos ese año.

Para entonces, John había reunido el coraje de preguntar a algunas empresas si querían que sus productos fueran compatibles con la tecnología. John no necesitaba ningún programa para el Atari. Generalmente decían que no. Entonces llegó al stand alquilado por On-Line Systems. Alguien le presentó a Ken Williams, que parecía simpático, y John le dijo a Ken que era un programador de negocios en lenguaje ensamblador, pero que estaba un poco harto de eso.

En ese momento Ken Williams había descubierto que la gente que podía escribir buenos juegos en lenguaje ensamblador era algo raro de encontrar. Quería atraer a estos programadores en lenguaje ensamblador a Coarsegold, California. On-Line Systems había experimentado un crecimiento

explosivo: en la última Feria de Computadoras, Ken había estado tanteando el terreno para Mystery House, y un año después era un editor de juegos establecido que necesitaba productos. Había puesto un anuncio en Softalk titulado “Se buscan autores”, prometiendo “las regalías más altas de la industria... No tendría que volver a trabajar en el horario de nadie más”. El anuncio mencionaba otro beneficio: la oportunidad de trabajar con el gurú de Apple, Ken Williams, que “estaría personalmente disponible en cualquier momento para discusiones técnicas, ayudando a depurar, a generar ideas...”. Ken era lo bastante inteligente como para darse cuenta de que los programadores que crearían estos productos no eran necesariamente trabajadores informáticos veteranos. Bien podrían ser adolescentes torpes. Como John Harris.

“Bueno”, le dijo Ken a John Harris, sin perder el ritmo, “¿qué te parecería programar entre los árboles?”.

Por muy atractivo que sonara, significaba trabajar para On-Line Systems, de la que John Harris sabía un poco. Sabía que vendían principalmente software de Apple. “No conozco el sistema de Apple”, dijo, omitiendo con tacto que, en la medida en que conocía el sistema de Apple, quería tirarlo por el inodoro.

Ken dijo las palabras mágicas. “Queremos expandirnos al sistema Atari. Simplemente no hemos encontrado a nadie que pueda programarlo”.

John estaba casi sin palabras.

“¿Puedes programarlo?”, preguntó Ken.

En un mes, Ken Williams le había comprado a John Harris un boleto de avión a Fresno, donde lo recogieron en el aeropuerto y lo llevaron por la Ruta 41 hasta Oakhurst. Ken le prometió a Harris un lugar para vivir y luego comenzaron a hablar del salario. John acababa de recibir un aumento en Gamma, por lo que los mil dólares mensuales que Ken le ofrecía en realidad habrían sido un recorte salarial. John encontró el coraje para decir que ahora estaba recibiendo más que eso. ¿Acaso Ken creía que podía pagar mil doscientos dólares al mes y agregarle el lugar gratis para vivir? Ken miró a Roberta (en ese momento, cualquier empleado de la pequeña oficina en línea podía en cualquier momento mirar por encima del hombro a cualquier otra persona que trabajara allí) y ella dijo que no creía que pudieran permitirse eso.

Williams dijo: "Te diré una cosa: ¿Qué tal si te ofrezco un treinta por ciento de regalías y no tienes que trabajar con la empresa? Trabajas desde tu casa y te daré setecientos dólares al mes para vivir hasta que termines tu primer juego, en dos o tres meses. Si no tienes un juego terminado para ese momento, no tendrás éxito en este negocio de todos modos”.

John pensó que eso sonaba genial. Sin embargo, cuando llegó a casa, su padre le dijo que se estaban aprovechando de él. ¿Por qué no obtener un salario mayor y un porcentaje menor de regalías? ¿Qué seguridad tenía John? John, que se había sentido intimidado por el tempestuoso Ken, no quería poner en peligro su oportunidad de vivir en una atmósfera construida alrededor de la piratería de juegos. Realmente quería salir de San Diego, piratear juegos y ser feliz. Aunque pudiera significar menos dinero, se aferraría a las regalías del treinta por ciento.

Fue la decisión más lucrativa que jamás tomó.

• • • • • • • •

Ken Williams había comprado varias casas en los alrededores de Oakhurst para beneficio de sus programadores. John Harris se mudó a una llamada Hexagon House, llamada así por la forma del piso superior, que era la única parte visible desde la calle: sobresalía del resto de la casa como un gran cenador sólido. Desde la puerta principal, se veían la sala de estar y la cocina; los dormitorios estaban en la planta baja. Allí vivía con John el hermano de veinte años de Ken, John Williams, que dirigía la división de publicidad y marketing de On-Line. Aunque a John Williams le gustaba Harris, lo consideraba un nerd.

El primer proyecto que John Harris le había mencionado a Ken estaba inspirado en el juego de arcade Pac-Man. Este fue el juego de monedas más popular en 1981, y pronto sería conocido como el más popular de todos los tiempos. John Harris no veía nada malo en ir a la sala de juegos, aprender a jugar y escribir su propia versión para que funcionara en el Atari 800. Para un hacker, traducir un programa útil o divertido de una máquina a otra era inherentemente bueno. La idea de que alguien pudiera tener Pac-Man, ese ingenioso jueguito en el que los fantasmas persiguen al comecocos amarillo que come puntos, aparentemente no era una consideración relevante para John Harris. Lo que sí era relevante era que el juego Pac-Man parecía encajar de forma natural con las características del Atari. Así que, aunque personalmente prefería los juegos con escenarios espaciales y muchos disparos, John le sugirió a Ken que hiciera un Pac-Man para el Atari 800.

Ken ya había estado comercializando un Pac-Man parecido para Apple bajo el nombre de Gobbler. El programa había sido escrito por un programador científico profesional llamado Olaf Lubeck, que le había enviado el juego a Williams, sin que él se lo pidiera, después de ver el anuncio de “Se buscan autores”. El programa vendía alrededor de ochocientas copias al mes, y Ken había acordado con Lubeck duplicarlo para el ordenador doméstico Atari.

Sin embargo, John Harris estaba horrorizado con el juego de Apple. “No se veía espectacular, no había animación”, explicó más tarde. “La detección de colisiones es muy implacable”. Harris no quería que Olaf agravara el error en su amado Atari traduciendo el juego de Apple bit a bit en el chip 6502, que Apple y Atari compartían. Esto significaría que no se utilizaría ninguna de las características que John consideraba superiores de Atari, la mayoría de las cuales estaban alojadas en chips separados. Sólo pensarlo le resultaba horroroso.

John insistió en que podría hacer un juego con mejor aspecto en un mes, y Ken Williams sacó a Lubeck del proyecto. John Harris se embarcó en un período de intensa hacking, que a menudo terminaba hasta la mañana siguiente. El estilo de John era despreocupado. John improvisaba. “Lo que sea que esté haciendo mi mente, simplemente la dejo fluir… las cosas salen bastante creativas”, explicó más tarde. A veces John podía ser sensible a esto, particularmente en momentos en que un programador más tradicional, armado con diagramas de flujo e ideas sobre la estructura estándar y la documentación clara, examinaba su código. Cuando John dejó Gamma Scientific para mudarse a Coarsegold, por ejemplo, le preocupaba que su reemplazo fuera alguien así, que tiraría todo su código inteligente, reemplazándolo por algo estructurado, conciso… y peor. Resultó que Gamma consideró a seis programadores, cinco de los cuales “tenían títulos que les salían por las orejas”, dijo John más tarde. El sexto era un hacker sin título; John rogó a sus jefes que contrataran al hacker.

“Pero él quiere tanto dinero como la gente que tiene título”, le dijo el jefe a John.

John dijo: “Él vale más”. Su jefe escuchó. Cuando John familiarizó a este nuevo empleado y le explicó su sistema, el nuevo hacker se emocionó mucho con el código de John. “¡Tú programas como yo!” dijo. “¡No pensé que hubiera alguien en el mundo que hiciera esto!”

Trabajando con grandes bloques conceptuales y manteniéndose concentrado, John tuvo un juego estilo Pac-Man funcionando en el Atari en un mes. Pudo usar algunas de las subrutinas que había desarrollado en esfuerzos anteriores. Este fue un buen ejemplo del tipo de crecimiento que la copia creativa podía fomentar: una especie de reencarnación de subrutina en la que un programador desarrollaba herramientas que trascendían ampliamente el rol de meras funciones derivadas. Un día, las subrutinas de John serían modificadas y utilizadas de una forma aún más espectacular. Esto constituia un resultado natural y saludable de la aplicación de los principios de los hackers. Fue una lástima que esta tercera generación de hackers tuviera que escribir sus propios kits de herramientas de software, complementándolos sólo con añadidos aleatorios de grupos de usuarios y amigos.

El juego Pac-Man se parecía mucho a la versión arcade. Bien podría haber sido uno de los mejores programas en lenguaje ensamblador escritos hasta el momento para el ordenador doméstico Atari. Pero cuando Harris llevó su trabajo a Ken Williams, hubo un problema. Últimamente, algunas empresas insistían en que los derechos de autor que poseían sobre los juegos que funcionaban con monedas hacían ilegales las traducciones no autorizadas para ordenadores domésticos. Uno de los mayores propietarios de derechos de autor era Atari, y envió la siguiente carta a pequeños editores como Brøderbund, Sirius y On-Line:

> PIRATERÍA DE SOFTWARE DE ATARI: ESTE JUEGO HA TERMINADO
>
> Atari es líder en el desarrollo de juegos como Asteroids™ y MISSILE COMMAND™... Agradecemos la respuesta que hemos recibido de los videófilos de todo el mundo que han hecho que nuestros juegos sean tan populares. Sin embargo, desafortunadamente, hay compañías e individuos que han copiado los juegos de ATARI en un intento de obtener ganancias inmerecidas de juegos que no desarrollaron. ATARI debe proteger nuestra inversión para que podamos seguir invirtiendo en el desarrollo de juegos nuevos y mejores. En consecuencia, ATARI advierte tanto a los piratas intencionales como a los individuos que simplemente desconocen las leyes de derechos de autor que ATARI registra las obras audiovisuales asociadas con sus juegos en la Biblioteca del Congreso y considera que sus juegos son de propiedad exclusiva. ATARI protegerá sus derechos haciendo cumplir enérgicamente estos derechos de autor y tomando las medidas adecuadas contra las entidades no autorizadas que reproduzcan o adapten copias sustanciales de los juegos de ATARI, independientemente del ordenador u otro aparato que se utilice en su ejecución...

Ken Williams sabía que Atari había gastado millones de dólares en los derechos de Pac-Man. Después de ver la duplicación del juego arcade de John Harris, con colores brillantes, movimientos rápidos y sin parpadeos, se dio cuenta de que era una copia tan fiel que no era comercializable. "Se parece demasiado a Pac-Man", dijo. "Has perdido el tiempo, John Harris". Sugirió que John modificara el juego. Harris se llevó el juego a casa y reprogramó los gráficos. Esta nueva versión era prácticamente la misma; la diferencia era que los fantasmas, esas pequeñas formas tontas que perseguían a Pac-Man, llevaban pequeños bigotes y gafas de sol. ¡Fantasmas de incógnito! Un comentario irónico perfecto sobre la estupidez de la situación.

No era exactamente lo que Ken Williams tenía en mente. Durante las dos semanas siguientes, John y Ken consultaron con abogados. ¿Cómo podían mantener la esencia de Pac-Man y al mismo

tiempo mantener a raya a Atari? Los abogados dijeron que lo único que Atari poseía realmente era la imagen del personaje, el aspecto del juego.

Así que se desarrolló un nuevo escenario, con el tema poco probable de la odontología preventiva. El hermano de Ken, John Williams, sugirió que los fantasmas se reemplazaran por "caras felices". Girarían y darían vueltas. John Harris reemplazó al Pac-Man amarillo por una dentadura postiza que hacía clic. En lugar de puntos, John dibujó "salvavidas" y programó una rutina que se produciría cuando el jugador eliminara los puntos: aparecería un cepillo de dientes que cepillaría los dientes. Nada de esto fue difícil de programar. John Harris simplemente dibujó las nuevas imágenes en tablas de formas y las escribió en su máquina existente. Una de las cosas maravillosas del ordenador era que se podía cambiar el mundo por impulso.

Los abogados aseguraron a Ken que este nuevo escenario de Jawbreaker no presentaba ningún problema con Atari. No conocían a Atari. Era una empresa propiedad del conglomerado Warner Entertainment; estaba dirigida por un ex ejecutivo textil que veía poca distinción entre el software de ordenador y cualquier otro artículo de consumo. Desde que los ingenieros ya no dirigían Atari, la empresa se había caracterizado por una burocracia que sofocaba los impulsos de los hackers. Los programadores de Atari cobraban mucho menos de lo que parecían exigir las astronómicas cifras de ventas de sus juegos, y convencer a los “expertos” en marketing para que lanzaran un juego innovador era una tarea formidable. Atari no incluía el nombre del programador del juego en el paquete; incluso se negaba a dar crédito a este artista cuando la prensa solicitaba el nombre de un autor. Cuando algunos de los mejores programadores de la empresa se quejaron, el exalumno textil que dirigía Atari supuestamente llamó a los hackers “diseñadores de toallas”: esos hackers estaban entre los muchos que renunciaron para formar empresas que diezmarían la cuota de mercado de Atari en cartuchos de juego.

Atari no pareció abordar esta pérdida de forma directa, sino que en cambio centró sus esfuerzos creativos en litigios y licencias de alto riesgo de propiedades aparentemente a prueba de fallos de otros medios, desde juegos que funcionaban con monedas hasta películas. Un ejemplo destacado fue Pac-Man, por el que Atari gastó millones. La idea era convertir primero el juego a la máquina de juegos VCS, luego a la línea de computadoras hogareñas de Atari, los 400 y 800. Las dos divisiones eran independientes y competían entre sí, pero ambos compartían el problema de la desaparición de programadores. Imaginemos, pues, la alegría de los ejecutivos de la división de ordenadores domésticos de Atari cuando un día, de repente, una persona cualquiera envió a Atari una copia de un programa que había estado circulando entre los grupos de usuarios aquel verano de 1981. Era una versión brillante de Pac-Man que funcionaba a la perfección en el Atari 800.

Fue el resultado de un clásico error de John Harris en el mundo real. Cuando John estaba trabajando en la revisión de Jawbreaker, algunas personas de una tienda de ordenadores de Fresno oyeron rumores de un Pac-Man brillante hackeado por el chico flacucho y nervioso que a menudo pasaba por allí para comprobar los periféricos y el software. Le pidieron a John Harris que les mostrara el juego. Sin pensar en restricciones que no son propias de los hackers, como el secreto corporativo, John Harris condujo hasta allí y los observó orgulloso mientras jugaban a la versión en progreso, y no vio nada inusual en su petición de pedirle prestada una copia del disco. Dejó una copia allí, regresó a la Hexagon House y continuó escribiendo su revisión.

Las copias del juego comenzaron a circular entre los grupos de usuarios de todo Estados Unidos. Cuando llegó a Atari, la gente de allí llamó a todas las empresas de software que se les ocurrieron

para encontrar a su autor. Finalmente, hablaron con Ken Williams, quien más tarde recordó que un ejecutivo de Atari le dijo que tenía un juego de Pac-Man de una calidad obviamente superior y que estaba buscando a su creador.

"Háblame del juego", dijo Ken, y el hombre de Atari lo describió como con caras felices. "¡Ese es John Harris!", dijo Ken. El hombre de Atari dijo que quería comprarle el programa a John Harris. Ken hizo que John Harris hiciera una llamada al jefe de adquisiciones de Atari, Fred Thorlin, desde la oficina de Ken. Según Ken, Thorlin estaba loco por el juego de John Harris. Le prometió a Harris una gran regalía, mencionó un concurso que Atari estaba organizando para el mejor programa de software, con un premio de veinticinco mil dólares, y dijo que ninguno de los participantes hasta el momento se había acercado al juego de Harris.

Pero John Harris recordó lo mezquino que había sido Atari cuando intentaba aprender lenguaje ensamblador. Sabía que había sido la carta de Atari a On-Line la que lo había obligado a hacer toda esa revisión en primer lugar. Atari había actuado, dijo John más tarde, como "un grupo de bebés", aferrándose a la información como un niño egoísta que protege un juguete de sus compañeros de juego. John Harris le dijo a Ken que no consideraría tener su nombre en nada publicado por Atari (no es que Atari hubiera mencionado poner su nombre en el programa), y que terminaría Jawbreaker para Ken.

Jawbreaker fue un éxito de ventas instantáneo. Casi todos los que lo vieron lo consideraron un hito para el ordenador doméstico Atari. Excepto Atari. Los hombres que dirigían Atari pensaban que el programa de John infringía su derecho, como propietarios de Pac-Man, de ganar tanto dinero como pudieran con el juego, comercializándolo de cualquier forma que consideraran conveniente. Si Ken Williams lanzaba un juego que daba al jugador la sensación de estar jugando a Pac-Man, especialmente si la versión de John Harris era mejor que la que pudiera idear el programador de Atari, ese jugador no sería propenso a comprar la versión de Pac-Man de Atari. Y Atari pensaba que su compra de la licencia de Pac-Man le daba derecho a cada centavo que se ganara con los juegos de ordenador domésticos que se jugaban como Pac-Man.

Era un desafío a la ética hacker. ¿Por qué Atari no debería estar contenta con una regalía pagada por personas que querían hackear el código de Pac-Man y, con el tiempo, mejorar el juego? ¿Se beneficiaba el público de que una empresa "poseyera" un software e impidiera que otros lo hicieran más útil?

Atari no veía mérito en ese argumento. Éste era el mundo real. Así que después del lanzamiento de Jawbreaker, Atari comenzó a presionar a On-Line Systems. Por un lado, quería que Ken Williams dejara de comercializar el juego. Por otro lado, quería comprar el programa de John Harris.

• • • • • • •

Ken no tenía ningún deseo de luchar contra Atari. No era un partidario incondicional de la ética hacker, así que no tenía ningún problema político - como lo tenía John Harris - en vender el programa a Atari. Cuando Fred Thorlin de Atari invitó a Ken y John Harris a que fueran a Sunnyvale, Ken aceptó.

John Harris, que rara vez parecía manejar la mecánica simple de la vida con tanta maestría como evocaba la magia de las entrañas del Atari 800, perdió su vuelo y llegó al complejo de edificios bajos de vidrio y hormigón de Atari en Sunnyvale después de que terminara la reunión. Había tenido suerte.

Ken contó más tarde la experiencia bajo juramento. Fred Thorlin lo había acompañado a una oficina donde lo esperaban algunos de los abogados internos de Atari. El consejero general asociado de Atari, Ken Nussbacher (que no estaba en la reunión), describió más tarde el enfoque de su empresa hacia editores como On-Line como “la zanahoria y el palo”, y este podría haber sido un ejemplo clásico. Según Ken Williams, un abogado les dijo que le gustaría que On-Line aceptara producir un juego de Pac-Man para Atari para que pudieran resolver tranquilamente el problema del infringir derechos que había creado el Jawbreaker (“la zanahoria”). Ken dijo que estaría encantado de tratar con Atari y que esperaba escuchar una propuesta.

Un segundo abogado le dio “el palo”. Según Ken, este abogado comenzó a gritar y a maldecir. Ken recordó que dijo “Atari lo había contratado para encontrar compañías que infringieran los derechos de autor de Atari y sacarlas del negocio... dijo que [Atari] podría pagar mucho, mucho más apoyo legal que yo y que si no cooperaba con ellos, me iban a sacar del negocio”.

Ken estaba tan asustado que temblaba. Pero les dijo a los abogados que un juez podría estar mejor calificado para ver si Jawbreaker era una infracción de derechos de autor.

En ese momento, Fred Thorlin le pidió al abogado que se calmara y considerara la posibilidad de que las dos compañías trabajaran juntas (“la zanahoria”). Hablaron sobre cuánto tiempo le llevaría a John Harris - el hacker de diecinueve años que amaba los ordenadores Atari pero los despreciaba y estaba perdido en algún lugar entre Coarsegold y Sunnyvale - terminar un nuevo juego de Pac-Man para Atari. Pero la oferta de Thorlin de un cinco por ciento de regalías era insultantemente baja. Después de que Thorlin le dijera "No tienes elección", el miedo de Ken comenzó a convertirse en ira. Decidió que prefería dejar que Atari lo demandara antes que ceder al chantaje. Para expresar su disgusto, arrojó las especificaciones para convertir Pac-Man sobre el escritorio de Thorlin y regresó a Coarsegold sin un acuerdo.

Durante un tiempo pareció que Atari cerraría On-Line. El hermano de Ken, John, recordó más tarde que un día alguien le hizo saber que Atari había obtenido una orden judicial para confiscar cualquier maquinaria que pudiera copiar discos de Jawbreaker: todos los ordenadores y unidades de disco de la empresa. El alguacil de Fresno estaba de camino. John Williams - que tenía veinte años y dirigía la empresa ese día - no pudo localizar a Ken y Roberta, así que ordenó a todos que se llevaran los ordenadores antes de que llegara el alguacil. De lo contrario, la empresa no habría podido seguir funcionando ni un día más.

Al Tommervik condujo un Toyota que resoplaba toda la noche para llegar al juzgado para estar junto a Ken durante la audiencia de la orden judicial, y le sugirió que Roberta le enviara por correo todos los originales para que los guardara. Dijo que encontraría un lugar para On-Line si Atari cerraba sus oficinas. Nunca se llegó a eso, pero hubo momentos muy tensos en el otoño de 1981.

John Harris estaba particularmente afectado. Había estado recibiendo lo suficiente en regalías como para comprarse una casa en las afueras de Oakhurst, una gran estructura de madera de color naranja. También se compró una camioneta con tracción en las cuatro ruedas. Estaba trabajando en un nuevo juego para On-Line, otro juego de laberinto llamado Mouskattack. A pesar de este repunte en su fortuna, John Harris estaba muy nervioso cuando se presentó a declarar a principios de diciembre.

Fue una imagen extraña. John Harris, un hacker de diecinueve años con vaqueros y camiseta, se enfrenta al mejor talento legal de uno de los mayores conglomerados de entretenimiento de Estados Unidos. El equipo legal de On-Line estaba encabezado por un tal Vic Sepulveda, un abogado de Fresno de pelo corto y gris, con unas grandes gafas negras de estilo aviador y una confianza

relajada. Su experiencia previa en derecho de propiedad intelectual había sido en un caso en el que algunos impresores habían insistido en que el texto de la homilía “Desiderata” era de dominio público.

Durante la declaración, John Harris estaba tan nervioso que no podía quedarse quieto. El abogado de Atari empezó preguntándole sobre sus primeros esfuerzos como programador, su trabajo en San Diego, cómo conoció a Ken, cómo escribió Jawbreaker... todas preguntas que John podía responder fácilmente, pero debido a su tensión se enredaba y se corregía a sí mismo; en un momento dado, se interrumpió y dijo: “Oh, Dios, eso sonó extraño”. John era una persona a la que normalmente le gustaba hablar de su trabajo, pero esto era diferente. Sabía que el objetivo de este abogado era hacerle decir algo que no quería decir, hacerle tropezar. Supuestamente una declaración es una búsqueda de la verdad, donde se hacen las preguntas más efectivas para obtener las respuestas más precisas. Debería funcionar como un programa fluido en lenguaje ensamblador, donde se dan las menores instrucciones posibles para acceder al chip 6502, dirigir la información dentro y fuera de la memoria, mantener las banderas adecuadas en los registros y, de entre miles de operaciones que se realizan cada segundo, obtener el resultado en la pantalla. En el mundo real no funcionaba así. La verdad que se encontraba en un ordenador no valía nada aquí. Era como si el abogado estuviera alimentando a John Harris con datos falsos con la esperanza de que el sistema colapsara.

Mientras que el hacker que había en John Harris estaba horrorizado por la naturaleza adversarial del sistema legal, el sistema legal tenía sus dificultades para adaptarse a él. Las reglas de la evidencia eran algo más rigurosas que los propios estándares de archivo de John. Ken Williams, en su propia declaración, había advertido de ello a los abogados de Atari cuando le preguntaron sobre el estado del código fuente de Harris para el programa y él había respondido: “Conozco a John Harris y estoy seguro de que no hay nada escrito. Él no trabaja así”.

¿No trabaja así? ¡Imposible! Un programador en Atari, como cualquier programador “profesional”, probablemente tenían que presentar código regularmente, lo que permitía una supervisión adecuada. Lo que los abogados de Atari no se dieron cuenta fue que Ed Roberts, Steve Wozniak e incluso los diseñadores de su propio Atari 800 habían creado una Tercera generación de hackers, idiotas expertos en microprocesadores, niños que no sabían ni un diagrama de flujo de Shinola, pero que podían usar un teclado como una paleta y abrirse camino hasta alcanzar cimas al estilo de Picasso.

> ABOGADO DE ATARI (a Ken): ¿No es un hecho que, por lo general, el programador que diseña estos juegos al menos produce un diagrama de flujo y luego escribe el código fuente manualmente antes de introducirlo?
>
> KEN WILLIAMS: No.
>
> ABOGADO DE ATARI: ¿Se sientan simplemente frente al teclado e introducen el programa?
>
> KEN WILLIAMS: Mis programadores suelen ser demasiado perezosos para hacer cualquier tipo de diagrama de flujo. En la mayoría de los casos, ni siquiera saben a dónde van cuando inician un programa. Intentan conseguir que funcione una rutina para ponerla en segundo plano y, a partir de ahí, pasar a algún juego.

No debió ser una gran sorpresa para los abogados de Atari, el segundo día de la declaración de John Harris, que no pudo encontrar la copia del juego Pac-Man anterior a Jawbreaker que había escrito. Las máquinas Atari de On-Line estaban en uso para copiar Wizard and the Princess, y el equipo de

John estaba roto, por lo que ni siquiera pudo encontrar el disco en el que estaba. "No está etiquetado en el frente", explicó John, diciendo: "Por lo que sé, debería estar en algún lugar de mi biblioteca".

Así que los abogados de Atari continuaron con John Harris, investigando la diferencia entre las versiones de su juego. Y a medida que avanzaba el examen, la línea entre la libertad creativa y el plagio se volvía cada vez más difusa. Sí, John Harris copió conscientemente de Pac-Man al programar su juego. Pero algunas de las rutinas que utilizó fueron escritas antes de que viera Pac-Man. Como el Atari 800 era radicalmente diferente de la máquina arcade Pac-Man, ya que utilizaba chips diferentes y requería técnicas de programación diferentes, el código de John Harris no se parecía en nada al código de Atari. Era completamente original.

Aun así, su primer juego se parecía a Pac-Man, utilizando los personajes protegidos por derechos de autor. Pero Ken se había negado a comercializar esa versión y John había cambiado los personajes. Atari insistió en que este cambio era insuficiente. Atari hizo que su jefe de marketing viniera a explicar "la magia de Pac-Man" al juez, llamándolo "un juego con un hombrecito, un pequeño Pac-Man" que devora puntos y píldoras de poder, que le permiten "dar la vuelta a la situación" y perseguir a los duendes que lo han estado devorando. El responsable de marketing continuó diciendo que la "magia de Atari" residía en su compromiso de comprar los derechos de los juegos de arcade populares.

Vic Sepulveda insistió en que John Harris simplemente había tomado la idea de Pac-Man de Atari, y citó la ley que establece que las ideas no pueden ser objeto de derechos de autor. El escrito de Vic enumeraba lado a lado las diferencias entre Pac-Man y Jawbreaker. La respuesta de Atari fue que a pesar de las diferencias, el juego era Pac-Man. De todos los laberintos que John Harris podría haber elegido, señalaron los abogados de Atari, eligió el laberinto de Pac-Man. Según la propia admisión de On-Line, ¡simplemente habían realizado una cirugía estética en una copia virtual de Pac-Man!

Pero el juez se negó a conceder a Atari una audiencia preliminar para obligar a On-Line a detener la comercialización de Jawbreaker. Examinó los dos juegos, pensó que podía distinguir la diferencia y dictaminó que, a la espera de un juicio completo, On-Line debería poder seguir comercializando Jawbreaker. Los abogados de Atari parecían atónitos.

David había vencido temporalmente a Goliat. Aun así, Ken Williams no estaba tan entusiasmado con la decisión como cabría esperar, porque On-Line tenía sus propios juegos y sus propios derechos de autor. A Ken Williams le estaba quedando claro que, en el fondo de su corazón, se identificaba con el punto de vista de Atari mucho más de lo que le importaba la ética hacker. “Si esto abre la puerta a que otros programadores copien mi software”, le dijo a Al Tommervik inmediatamente después de la decisión, “lo que pasó aquí fue algo malo”. Llegaría a un acuerdo en la demanda antes de que llegara a juicio.

Capítulo 17. Campamento de verano

Ken Williams empezó a confiar en gente como John Harris, hackers de tercera generación influidos no tanto por Robert Heinlein o Doc Smith como por Galaxian, Dungeons and Dragons y Star Wars. Estaba floreciendo toda una subcultura de programadores-hackers creativos y diseñadores de juegos, más allá del alcance de los cazatalentos ejecutivos. La mayoría de ellos todavía estaban en la escuela secundaria.

Para atraer a los jóvenes programadores a Coarsegold, Williams publicó anuncios en Los Angeles Times tentándolos a "entrar en Yosemite". Una de las respuestas típicas fue la de un hombre que le dijo a Ken: "Mi hijo es un gran programador de Apple y le gustaría trabajar contigo". "¿Por qué no me dejas hablar con tu hijo?", preguntó Ken. El hombre le dijo que su hijo no se comportaba bien por teléfono. En la entrevista de trabajo en Oakhurst, el hombre insistió en responder todas las preguntas por su hijo, un chico rubio de dieciséis años, de ojos redondos y pelo de melocotón en las mejillas, que parecía intimidado por toda la situación. Nada de esto importó cuando Ken descubrió que el chico era capaz de captar las complejidades del lenguaje ensamblador de Apple. Ken lo contrató por tres dólares la hora.

Poco a poco, Ken Williams empezó a llenar la casa que compró en la zona de Sierra Sky Ranch, justo más allá de Oakhurst, donde la Ruta 41 comienza a ascender a alturas de más de cinco mil pies. Además del alquiler gratuito, estaban los tutoriales de gráficos improvisados de Ken. Ken ahora era conocido como un mago certificado de Apple. Podía activar su curiosidad de hacker casi por capricho. Se negaba a aceptar lo que otros consideraban limitaciones genéricas de Apple. Usaba técnicas de pasar páginas, de exclusión o enmascaramiento... cualquier cosa para que algo saliera en la pantalla. Cuando miraba el programa de otra persona, podía oler un problema, rodearlo, llegar al meollo del asunto y encontrar una solución.

La sede corporativa de On-Line en 1981 estaba en el segundo piso de una estructura de madera marrón oscuro en la Ruta 41, cuya planta baja albergaba una papelería y una pequeña imprenta. Se entraba a la oficina después de subir un tramo de escaleras en el exterior del edificio; había que salir más allá de la escalera para ir al baño. Dentro de la oficina había un grupo de escritorios, menos escritorios que empleados. La gente jugaba continuamente a las sillas musicales para reclamar espacio en el escritorio y el uso de uno de los varios Apple. Cajas de discos, monitores de computadora desechados y pilas de correspondencia estaban apiladas en el suelo. El desorden era alucinante. El nivel de ruido, rutinariamente intolerable. El código de vestimenta, inexistente. Era una anarquía productiva, reminiscente de la atmósfera desestructurada del laboratorio de inteligencia artificial o del Homebrew Club. Pero como también era un negocio próspero y los participantes eran tan jóvenes, la oficina de On-Line parecía una extraña combinación de Animal House y The Millonaire.

Era indicativo de las prioridades de Ken Williams. Estaba involucrado en un nuevo tipo de negocio en una industria completamente nueva y no estaba dispuesto a establecer el mismo ambiente odioso, claustrofóbico, secreto y burocrático que tanto despreciaba en casi todas las empresas en las que había trabajado. Él era el jefe, pero no sería el tipo de jefe que era Dick Sunderland en Informática, obsesionado con los detalles. Él tenía el control del panorama general. Además de hacerse rico, algo que parecía estar encajando perfectamente ya que sus programas se ubicaban regularmente entre los diez o quince primeros de la lista de los "30 más vendidos" que publicaba Softalk cada mes, Ken sentía que tenía una doble misión que cumplir en On-Line.

La primera era divertirse, un elemento que sentía que lamentablemente había faltado en los establecimientos de la Antigüedad, limitados por el decoro. Ken Williams se convirtió, en efecto, en el consejero principal de un campamento de verano de alta tecnología. Había diversión, alboroto, bebida y consumo de marihuana. Drogados o no, todos estaban en la cima, trabajando en un campo que les parecía bien, política y moralmente. La fiesta extendida se alimentaba regularmente con una afluencia de sobres con dinero.

También llegaban paquetes con juegos nuevos, ya fueran juegos de competidores amistosos como Sirius o Brøderbund, juegos de posibles superestrellas del software que buscaban ser publicados o juegos de uno de los autores externos de On-Line que trabajaban bajo la supervisión de Ken. No importaba. Todo se detenía para los juegos nuevos. Alguien hacía copias y todos se iban a los Apples, jugaban al juego, se burlaban de sus errores, admiraban sus características y veían quién podía obtener la puntuación más alta. Mientras el dinero siguiera llegando, y ciertamente lo hacía, ¿a quién le importaba un poco de desorganización o una tendencia excesiva a pasar al modo fiesta?

Los forasteros visitaban la oficina y no creían lo que veían. Jeff Stephenson, por ejemplo. A sus treinta años, era un programador experimentado que había trabajado recientemente para Software Arts, la empresa de Cambridge, Massachusetts, que había escrito el programa más vendido de todos los tiempos para Apple, la «hoja de cálculo» financiera VisiCalc. Esa empresa también estaba dirigida por programadores: Jeff podía recordar a los dos presidentes, uno de ellos un ex hacker del MIT y el otro un meticuloso y joven judío ortodoxo, que discutieron durante media hora sobre dónde debería ir una coma en un informe. Jeff, un vegetariano tranquilo y modesto que tenía cinturón negro en esgrima coreana, se había mudado a las montañas con su esposa recientemente y llamó a On-Line para ver si la empresa más cercana a su nuevo hogar necesitaba un programador. Se puso unos vaqueros de pana y una camisa deportiva para la entrevista; su esposa le sugirió que se vistiera más elegante. “Esto es la montaña”, le recordó Jeff, y condujo por Deadwood Mountain hasta On-Line Systems. Cuando llegó, Ken le dijo: “No sé si vas a encajar aquí, pareces un poco conservador”. Contrató a Jeff de todos modos, por dieciocho mil dólares al año, once mil menos de lo que ganaba en Software Arts.

En ese momento, el proyecto más ambicioso que On-Line había intentado jamás estaba empantanado en un desastre organizativo. Time Zone, el juego de aventuras en el que Roberta había estado trabajando durante casi un año, era un programa fuera de control, dominado por un equivalente literario de Creeping Feature Creature. Casi ebria de la ambición vertiginosa de crear en el ordenador, Roberta estaba tramando un escenario que no sólo recrearía escenas de todo el mundo, sino que abarcaría la amplitud de la historia registrada, desde el amanecer del hombre hasta el año 4081. Cuando Roberta jugaba a un buen juego de aventuras, siempre deseaba que nunca terminara; este juego, decidió, tendría tantas tramas y habitaciones que incluso un aventurero experimentado tardaría un año en resolverlo. Verías la caída de César, sufrirías las guerras napoleónicas, lucharías contra guerreros samuráis, rapearías con aborígenes australianos prehistóricos, navegarías con Colón, visitarías cientos de lugares y serías testigo de todo el panorama de la experiencia humana, para finalmente terminar en el planeta Neburon, donde el malvado líder Ramadu planea destruir la Tierra. Una epopeya de microordenador, concebida por una ama de casa en el centro de California.

La programación de este monstruo estaba paralizando el negocio de On-Line. Un programador del personal estaba trabajando en una rutina para triplicar la velocidad con la que el programa rellenaría los colores de las imágenes de alta resolución. El joven programador cuyo padre había conseguido su empleo trataba de lidiar con la lógica del juego, mientras que un ex alcohólico que se había

ganado el título de programador tecleaba los mensajes del lenguaje de desarrollo de aventuras. Un adolescente local dibujaba minuciosamente las mil cuatrocientas imágenes, primero en papel cuadriculado y luego las volvía a trazar en una tableta gráfica de Apple.

Se le pidió a Jeff Stephenson que uniera de alguna manera el programa. Estaba consternado por la desorganización y horrorizado por la fecha límite: otoño, por lo que el juego estaría a la venta para Navidad. (Más tarde llegaría a la conclusión de que cualquier fecha límite que daba Ken era generalmente demasiado optimista por un factor de tres.)

A pesar de que el proyecto iba tan retrasado con respecto a lo previsto, la empresa seguía funcionando como un campamento de verano. El martes por la noche era la "Noche de los hombres", y Ken había salido a beber. Todos los miércoles, la mayoría del personal se tomaba el día libre para ir a esquiar a Badger Pass, en Yosemite. Los viernes al mediodía, On-Line celebraba un ritual llamado “Breaking Out the Steel”. “Steel” era el aguardiente de menta de Steel, claro pero potente, que era la bebida preferida de On-Line Systems. En la jerga de la empresa, una gran cantidad de acero te “dejaba en un estado de trineo”. Una vez que sacaban el acero los viernes, se podía suponer razonablemente que el trabajo en Time Zone se detendría mientras el personal, con Ken a la cabeza, exploraba la zona nebulosa y atemporal del trineo.

Llegó y pasó la Navidad, y Time Zone no se comercializó hasta febrero. Doce veces el tamaño de *El mago y la princesa*, llenando ambas caras de seis disquetes, se vendía al por menor por cien dólares. El primero en resolverlo, un jovial fanático de los juegos de aventuras llamado Roe Adams (que también era el principal crítico de Softalk), estuvo prácticamente sin dormir durante una semana hasta que derrotó a Ramadu y declaró que la creación de Roberta era una de las mayores hazañas de la historia de los juegos.

Time Zone, sin embargo, no se ganó ni de lejos la notoriedad de otra aventura en línea que estaba en sintonía con el espíritu de la empresa. El juego se llamaba *Softporn*. En la primavera de 1981, Ken había conocido a un programador que había estado hablando con editores sobre un juego de aventuras que había escrito y que estaba tratando, con poco éxito, de promocionarse. Este juego no era la aventura habitual en la que buscas joyas, o intentas resolver un asesinato, o intentas derrocar a un malvado Emperador Nyquill del Planeta Yvonne. En este juego, eras un soltero cuya misión era encontrar y seducir a tres mujeres. El programador había escrito el programa como un ejercicio de entrenamiento para aprender sobre bases de datos, utilizando el tema sexual para hacerlo interesante. Era el tipo de cosas que los hackers, al menos los que sabían que existía algo llamado sexo, habían estado haciendo durante años, y era raro encontrar un centro informático sin su propio especial sexual, ya fuera un generador de chistes obscenos o un programa para imprimir o mostrar en pantalla alguna mujer desnuda. La diferencia era que en 1981, todo tipo de cosas que los hackers habían estado haciendo como fanfarroneadas técnicas cósmicas tenían un valor de mercado repentino en las traducciones para computadoras domésticas.

El programa en cuestión era una variación mejorada del original. Se volvería vil solo si usabas obscenidades en tus órdenes. Aún así, para ganar el juego necesitabas tener sexo con una prostituta, comprar un condón para evitar enfermedades venéreas y practicar sadomasoquismo con una rubia que insistía en casarse contigo antes de poder acostarte con ella. Si querías tener éxito en esta aventura, las respuestas que escribieras en la computadora tenían que ser imaginativamente seductoras. Pero había peligros: si te encontrabas con la "rubia voluptuosa" y escribías COMER RUBIA, la computadora escribiría un pasaje insinuando que la rubia estaba inclinada sobre ti y

practicándote sexo oral. ¡Pero luego te mostraría sus dientes relucientes y te lo arrancaría de un mordisco!

Para aquellos con sentido del humor resistente a ese tipo de cosas, Softporn era un juego de Apple excepcionalmente deseable. La mayoría de los editores de software no querían tener nada que ver con el juego; se consideraban empresas "familiares". Pero Ken Williams pensó que el juego era un desenfreno: se lo pasó genial resolviendo la aventura en tres o cuatro horas. Pensó que la controversia sería divertida. Aceptó comercializar Softporn.

Un día, no mucho después, Ken entró en la oficina y dijo: "¿Quién quiere venir a mi casa y tomarse fotos desnuda en el jacuzzi?".

La idea era conseguir que tres mujeres posaran en topless en el jacuzzi de Ken para el anuncio de Softporn. En algún lugar de la imagen habría un ordenador Apple, y en la bañera con las tres mujeres desnudas habría un camarero sirviéndoles bebidas. Pidieron prestado un camarero de The Broken Bit, un restaurante de carnes de Coarsegold que era el único lugar decente para comer en la ciudad. Las tres mujeres, todas ellas trabajadoras de On-Line, que se quitaron las blusas fueron la contable de la empresa, la esposa del asistente de Ken y Roberta Williams.

El anuncio a todo color, con las mujeres sosteniendo copas de vino (el agua del jacuzzi cubría sus pezones con mucho tacto), el camarero completamente vestido sosteniendo una bandeja con más copas de vino blanco y un ordenador Apple de pie, bastante desamparado, al fondo, causó sensación. On-Line recibió su cuota de mensajes de odio, algunos de ellos llenos de pasajes bíblicos y profecías sobre la condenación que se avecinaba. La historia del juego y el anuncio atrajeron la imaginación de los servicios de noticias, y la imagen se publicó en Time y en la cadena UPI.

A Ken Williams le encantaba la publicidad gratuita. Softporn se convirtió en uno de los productos más vendidos de On-Line. Las tiendas de ordenadores que lo querían se mostraban reacias a encargar sólo ese programa. Así que, como el adolescente que va a la farmacia y dice: "Me gustaría un peine, pasta de dientes, aspirina, aceite bronceador, una resma de papel de oficina y, oh, ya que estoy aquí, también podría comprarme este Playboy", los dueños de las tiendas encargaban una muestra completa de productos de On-Line... y también copias de Softporn. Ken supuso que Softporn y su efecto dominó duplicarían sus ingresos.

Divertirse, hacerse rico, hacerse famoso y organizar una fiesta interminable eran sólo una parte de la misión de Ken; también había un componente más serio. Estaba desarrollando una filosofía sobre la computadora personal y su capacidad para transformar la vida de las personas. Apple y el grupo de computadoras similares eran sorprendentes no sólo por lo que hacían, sino también por su accesibilidad. Ken había visto a gente totalmente ignorante sobre computadoras trabajar con ellas y ganar confianza, de modo que toda su perspectiva de la vida había cambiado. Al manipular un mundo dentro de una computadora, la gente se dio cuenta de que era capaz de hacer que las cosas sucedieran con su propia creatividad. Una vez que tenías ese poder, podías hacer cualquier cosa.

Ken Williams se dio cuenta de que podía exponer a la gente a ese tipo de transformación, y se propuso utilizar la empresa que él y Roberta habían fundado como una especie de proyecto de rehabilitación para algunas de las personas infrautilizadas de Oakhurst y Coarsegold.

La zona había estado sufriendo la recesión, especialmente en el ámbito industrial y minero que una vez la sustentaba. No había habido ningún auge desde la Fiebre del Oro. On-Line Systems se convirtió rápidamente en el mayor empleador de la zona. A pesar del estilo de gestión poco

ortodoxo de Ken, la aparición de una empresa de alta tecnología en la ciudad fue una bendición: formaban parte de una comunidad, les gustara o no. Ken disfrutaba de su papel de nuevo rico, padre de familia, y despachaba sus responsabilidades cívicas con su habitual tendencia a los excesos (por ejemplo, enormes donaciones al departamento de bomberos local). Pero los amigos íntimos que Ken y Roberta harían no parecían provenir de las capas superiores de la sociedad de Oakhurst. Eran, en cambio, las personas que Ken sacó de la oscuridad gracias al poder de la computadora.

El trabajo de Rick Davidson consistía en limpiar barcos con arena, y su esposa Sharon trabajaba como mucama en un motel. Ken los contrató a ambos; Rick acabó convirtiéndose en vicepresidente a cargo del desarrollo de productos, y Sharon dirigía el departamento de contabilidad. Larry Bain era un fontanero desempleado que se convirtió en el jefe de adquisición de productos de Ken.

Una transformación particularmente dramática ocurrió en la persona de Bob Davis. Era el espécimen principal en el laboratorio humano de Ken, On-Line Systems, una empresa misionera que usaba computadoras para transfigurar a los fracasados y a los que nunca fueron en maestros de la tecnología. A los veintisiete años, Davis era un ex músico y cocinero de comida rápida con cabello largo y pelirrojo y una barba descuidada. En 1981, estaba trabajando en una licorería. Estaba encantado con la oportunidad de reformar su vida mediante computadoras, y Ken estaba aún más encantado con la transformación. Además, la vena salvaje de Bob Davis parecía coincidir con una perversión similar en la personalidad de Ken.

Siempre que Ken Williams iba a la licorería a comprar su bebida, Bob Davis le rogaba que le diera trabajo. Davis había oído hablar de este nuevo tipo de empresa y sentía curiosidad por las computadoras. Ken finalmente le dio trabajo: copiar discos por la noche. Davis comenzó a ir durante el día para aprender programación. Aunque había abandonado la escuela secundaria, parecía tener una afinidad por BASIC y buscó ayuda adicional en el grupo de jóvenes hackers de Ken. Davis, astuto, vio que On-Line estaba ganando mucho dinero con esos juegos y se comprometió a escribir uno él mismo.

Bob y su esposa comenzaron a salir con los Williams. On-Line Systems era una empresa lo suficientemente flexible como para adaptarse a un acuerdo que desafiaba los tabúes tradicionales entre propietarios y empleados. Iban de viaje juntos, a lugares como Lake Tahoe. El estatus de Bob en la empresa aumentó. Lo nombraron programador y fue director de proyectos de la empresa Time Zone. La mayor parte del tiempo, tecleaba en código ADL, sin saber mucho sobre lenguaje ensamblador. A algunas personas les molestaba (incluso al amable Jeff Stephenson, a quien Bob le gustaba mucho) que Bob Davis anduviera por ahí llamándose programador, cuando un programador de verdad, cualquiera con credenciales de hacker, debería haber sido capaz de realizar una magia mucho más concentrada que la que tenía Davis.

Sin embargo, una vez que Davis aprendió las herramientas ADL de Ken, tuvo la clave para escribir un juego de aventuras de nivel profesional. Siempre le había interesado la mitología y leyó algunos clásicos griegos, en particular los que trataban sobre Jasón, e incorporó los cuentos antiguos en un juego de aventuras. Programó el juego, según afirmó, en su tiempo libre (aunque algunos en On-Line pensaron que descuidó sus obligaciones de zona horaria por su propio proyecto) y con algo de ayuda de Ken, lo terminó. Menos de un año después de ser rescatado de su trabajo como empleado en una licorería, era una estrella del software. El abogado de On-Line supuso que podría haber un problema en llamar al juego Jasón y el vellocino de oro porque ese era el título de una película que

podría estar protegida por derechos de autor, por lo que On-Line lanzó el juego como Ulises y el vellocino de oro.

Fue un éxito instantáneo, colocándose cómodamente en el Top Thirty de Softalk. La revista Videogame Illustrated lo calificó como “uno de los videojuegos más importantes y desafiantes jamás creados”, aunque en realidad no representó ningún avance significativo con respecto a las aventuras de alta resolución anteriores, excepto que era más largo y sus gráficos parecían considerablemente más artísticos que las imágenes de Mystery House con su aspecto de monigote. La revista también entrevistó a Davis, que parecía todo un experto, hablando de lo que los consumidores de juegos podrían esperar en los próximos cinco años (“ordenadores conectados a todos los teléfonos y televisores… síntesis de voz… reconocimiento de voz… efectos especiales generados por videodiscos…”). Un escenario utópico, ¿y por qué no? Miren lo que los ordenadores habían hecho por Bob Davis.

• • • • • • •

Los cambios que los ordenadores personales estaban provocando en la vida de las personas no se limitaban en absoluto a California. En todo el país, la computadora estaba abriendo nuevas áreas de creatividad. Parte del sueño de los hackers era que la computadora liberara a las personas que tenían tendencias creativas insatisfechas. Incluso podrían ascender a un nivel de magia que les permitiera ganarse el apelativo de hackers. Ken Williams ahora podía ver que esto estaba sucediendo. Casi como si estuviera predestinado, algunos de sus programadores, una vez inmersos en comunión con la máquina, habían florecido confiadamente. Ninguna transformación fue más dramática que la de Warren Schwader.

Tal vez el evento más significativo en la vida de Warren Schwader ocurrió en 1977, cuando Warren tenía dieciocho años: su hermano compró uno de los primeros ordenadores Apple II. Su hermano había quedado paralítico en un accidente de coche y quería la Apple para aliviar su aburrimiento. Warren, alto, rubio, de rasgos gruesos y de hablar lento, tuvo que ayudar a su hermano a introducir comandos en la Apple. Y fue Warren quien se convirtió en el hacker.

En ese momento, Warren trabajaba en la Parker Pen Company en su ciudad natal, en la zona rural de Wisconsin. Aunque Warren tenía talento para las matemáticas, dejó la escuela después de terminar la escuela secundaria. Su trabajo en Parker consistía en operar una máquina de moldeo por inyección, que consistía en un gran molde y un tubo donde se calentaba el plástico. El plástico caliente se inyectaba en el molde y, después de veinte segundos de enfriamiento, Warren abría la puerta y sacaba las piezas de bolígrafo recién formadas. Luego volvía a cerrar la puerta. Warren Schwader consideraba que el trabajo era un reto. Quería que las piezas del bolígrafo fueran perfectas. Constantemente ajustaba el cargador, giraba la llave o apretaba las tuercas y los tornillos de la moldeadora. Amaba esa máquina. Años después de dejar Parker, dijo con orgullo que las piezas del bolígrafo de su moldeadora eran, en efecto, perfectas.

Abordaba la programación con la misma compulsividad meticulosa. Todos los días probaba una demostración gráfica diferente. Por la mañana decidía qué quería probar. Durante los intervalos de veintidós segundos que le permitía su máquina de moldeado, usaba lápiz y papel para hacer un diagrama de flujo de un programa para la demostración. Por la noche, se sentaba en el Apple y depuraba el programa hasta que el efecto deseado llenaba la pantalla. Le gustaban especialmente las pantallas caleidoscópicas y multicolores.

Una de las demostraciones gráficas que Warren probó le atrajo tanto que decidió intentar convertirla en un juego. Desde que jugó por primera vez al Pong en los salones recreativos, Warren había sido un fanático de los videojuegos. Warren Schwader intentó copiar un juego que había visto en una sala de juegos: tenía una paleta en la parte inferior de la pantalla y pequeños ladrillos en la parte superior. Si golpeabas un punto con la paleta, rebotaba como una máquina de pinball. Eso le llevó a Warren un mes de intervalos de veinte segundos y depuración nocturna, y aunque estaba escrito en gráficos de baja resolución, que no eran tan nítidos como los que se podían hacer en lenguaje ensamblador y alta resolución, el juego que logró también era bueno.

Hasta ese momento, Warren había estado trabajando en el Apple únicamente para descubrir qué podía hacer con él. Se había absorbido en el proceso puro. Pero al ver estos juegos en la pantalla, juegos que había creado de la nada, juegos que podrían haber sido las cosas más creativas que había logrado en su vida, Warren Schwader comenzó a darse cuenta de que su computación podía realmente producir un resultado tangible. Como un juego que otros podrían disfrutar.

Esta epifanía llevó a Warren más profundamente a la máquina. Decidió hacer un juego en lenguaje ensamblador, incluso si le llevó meses. No había libros sobre el tema y, desde luego, nadie que Warren conociera en Wisconsin podía decirle nada al respecto.

Además, el único ensamblador que tenía Warren era el sencillo y lento miniensamblador que venía integrado en el Apple. Nada de esto detuvo a Warren Schwader, que en personalidad y perspectiva se parece mucho a la legendaria tortuga que finalmente supera a la liebre.

Warren hizo un juego en lenguaje ensamblador llamado *Smash-Up*, en el que el jugador, controlando un pequeño automóvil, intenta evitar colisiones frontales con otros automóviles. Consideró que era lo suficientemente bueno como para venderlo. Warren no tenía suficiente dinero para un anuncio en una revista, así que simplemente hizo tantas copias como pudo en cintas de casete y las envió a las tiendas de computadoras. Esto fue en 1980, cuando el recién creado mercado de juegos de Apple estaba cambiando del casete a los disquetes más rápidos y versátiles. Warren vendió solo unos dos mil dólares en el juego Smash-Up, invirtiendo casi el doble de esa cantidad en gastos.

La compañía Parker Pen cerró la fábrica, por lo que Warren tuvo mucho más tiempo para trabajar en su próximo juego. “Acababa de aprender a jugar a Cribbage y me encantaba”, recordaría más tarde Schwader. “No había nadie que supiera jugarlo conmigo, así que dije: ‘¿Por qué no escribo un programa que juegue a Cribbage?’”. Trabajó en él unas ochocientas horas, a menudo hasta el amanecer de Wisconsin. Intentaba hacer trucos gráficos que no entendía del todo, cosas que más tarde conocería como direccionamiento indirecto y gráficos de página cero. Trabajó tan duro en el juego que “todo el tiempo sentí que estaba dentro de la computadora. La gente me hablaba, pero yo no podía interactuar”, dijo más tarde. Su lengua materna ya no era el inglés, sino los jeroglíficos hexadecimales de LDX #$0, LDA STRING, X, JSR $FDF0, BYT $0, BNE LOOP.

El programa terminado era magnífico. Warren había desarrollado algunos algoritmos inspirados que permitían a la computadora evaluar su mano con doce reglas principales. Consideró que el programa era impecable en su elección de cartas para tirar a la cuna. Fue sólo porque Warren estaba familiarizado con las características del programa (lo conocía como un compañero de cartas de la vieja escuela) que pudo ganarle alrededor del sesenta por ciento de las veces.

Warren Schwader envió el juego a Ken Williams, quien quedó impresionado con la lógica y los gráficos, que ofrecían una imagen clara y nítida de cada carta repartida. Lo que era aún más asombroso era que Schwader había hecho esto en el miniensamblador limitado de Apple.

Era como si alguien le hubiera enviado a Ken una mecedora bellamente elaborada y luego le hubiera dicho que el artesano no había usado sierra, torno u otras herramientas convencionales, sino que había construido la silla con una navaja. Ken le preguntó a Warren si quería trabajar para On-Line. Vivir en el bosque. Entrar a Yosemite. Unirse al campamento de verano salvaje y alocado de una empresa de la nueva era.

Warren había estado subsistiendo con los doscientos dólares mensuales que recibía del estado por cuidar de su hermano. A Warren le preocupaba dejarlo con enfermeros de día, pero su hermano le dijo que este asunto de On-Line era una gran oportunidad y debía aprovecharla. Y a Warren le atraía la idea de salir a ganar dinero programando juegos y viviendo en el bosque. Así que decidió hacerlo. Pero había una parte del paquete que no le atraía: la diversión, el alboroto, el consumo de alcohol y el consumo de marihuana en el campamento de verano, que eran prácticas habituales en On-Line Systems.

Warren era testigo de Jehová.

Cuando Warren trabajaba en Cribbage, su madre había muerto. Warren empezó a pensar en su futuro y en cuál era su propósito en la vida. Descubrió que las computadoras eran lo principal por lo que vivía. Sintió que tenía que haber algo más y se dedicó a la religión de su difunta madre. Comenzó a estudiar intensamente la Biblia y juró que su nueva vida en California se caracterizaría por la adhesión a los preceptos de Jehová.

Al principio, esto no interfirió mucho en su vida en On-Line. Warren Schwader no criticó la dolce vita en On-Line Systems. Pero debido a los hábitos impíos de sus colegas, generalmente limitaba sus transacciones con ellos a discusiones técnicas o de negocios. Prefería quedarse con gente de su fe para estar protegido de la tentación.

Vivía solo, sin cobrar nada, en una de las casas de Ken, una pequeña casa de dos habitaciones. Su vida social se limitaba a un salón del Reino de los Testigos de Jehová en Ahwahnee, a ocho kilómetros al oeste de Oakhurst. La primera vez que fue a un servicio allí, sintió que había hecho más amigos que nunca antes. Aprobaban las computadoras, le decían que podían hacer mucho bien al hombre, aunque uno debe tener cuidado de que se puede hacer mucho daño a través de las computadoras. Warren se dio cuenta de que el amor que tenía por el hacking era una amenaza para su devoción a Dios, y aunque todavía amaba la programación, trató de moderar sus sesiones de hacking para no desviarse de su verdadero propósito. Así que, mientras seguía programando por las noches, también mantenía sus estudios bíblicos, y durante las tardes y los fines de semana viajaba por la zona, llamando a las puertas y entrando en las casas de la gente, llevando ejemplares de ¡Despertad! y La Atalaya, y predicando la fe de Jehová.

Mientras tanto, estaba trabajando en un juego basado en algunas de las subrutinas en lenguaje ensamblador más rápidas y espectaculares que Ken había creado hasta el momento. Era un juego como Space Invaders, en el que tenías un cohete y tenías que luchar contra oleadas de invasores. Pero las oleadas estaban llenas de formas extrañas y se movían en todo tipo de direcciones, y si el jugador intentaba enviar un flujo constante de balas para luchar contra ellas, su “pistola láser” se sobrecalentaría y se enfrentaría a una muerte casi segura. Era el tipo de juego diseñado para provocar un paro cardíaco en los débiles de corazón, tan feroces eran los atacantes y tan violentas

las explosiones. No fue exactamente un hito en los juegos de Apple, ya que derivaba de la escuela de juegos de disparos Space Invaders, pero sí representó una escalada en la pirotecnia gráfica y la intensidad de los juegos. El nombre de este programa de computadora era Threshold, y le hizo ganar a Warren Schwader casi cien mil dólares en regalías, un porcentaje significativo de los cuales fue diezmado al Kingdom Hall en Ahwahnee.

Pero a medida que Warren se acercaba a la comunidad del Reino, comenzó a cuestionar profundamente el tipo de cosas que había estado haciendo para On-Line. Se preguntó si su propia alegría en la programación no era una especie de pecado. El acto de programar el juego había sido carnal: Warren había trabajado toda la noche con su estéreo a todo volumen con Led Zeppelin (la banda de rock de Satanás). Peor aún, la naturaleza de disparos del juego no dejaba dudas de que glorificaba la guerra. El estudio de las escrituras de Warren lo convenció de que uno ya no debería aprender guerra. Se sentía avergonzado de que un juego de guerra que él había programado fuera jugado por niños.

Por eso no le sorprendió ver un artículo de ¡Despertad! sobre los videojuegos que los comparaba con las drogas y decía que los juegos bélicos “promueven la agresión sin piedad”. Warren decidió dejar de programar juegos violentos y juró que si Watchtower se pronunciaba firmemente en contra de todos los juegos, tendría que dejar de programar y buscar otra cosa que hacer con su vida.

Empezó a trabajar en un juego no violento con una temática circense. El trabajo fue lento porque trató de no perderse en la programación hasta el punto de convertirse en un zombi que hubiera perdido el contacto con Dios. Se deshizo de todos sus álbumes de hard rock y puso música como Cat Stevens, Toto y los Beatles. Incluso empezó a gustarle música que antes hubiera considerado cursi, como Olivia Newton-John (aunque cuando ponía su disco siempre tenía que acordarse de levantar la aguja cuando sonaba la canción pecaminosa Physical).

Aun así, cuando Warren habló de su nuevo juego, de cómo estaba usando animación de doble página con doce patrones diferentes para controlar los barriles rodantes sobre los que el personaje debía saltar, o de cómo no tendría parpadeos y sería “cien por ciento jugable”, estaba claro que, a pesar de sus esfuerzos estéticos, se enorgullecía sensualmente del hack. La programación significaba mucho para él. Había cambiado su vida, al darle poder, lo convirtió en alguien.

• • • • • • • •

A pesar de lo mucho que a John Harris le encantaba vivir lejos de San Diego, en las colinas de Sierra Nevada, de lo mucho que apreciaba la atmósfera relajada del campamento de verano y de lo feliz que estaba de que sus programas fueran reconocidos como esfuerzos coloridos y creativos, una parte crucial de su vida era totalmente insatisfactoria. Era una enfermedad común de los hackers de tercera generación, para quienes el hackeo era importante, pero no todo, como lo era para los hackers del MIT. John Harris ansiaba una novia.

Ken Williams se tomaba en serio las preocupaciones de sus jóvenes programadores. Un John Harris feliz sería un John Harris que escribiera juegos exitosos. Roberta Williams también sentía afecto por el ingenuo joven de veinte años y se sintió conmovida por lo que ella creía que era un amor secreto que él albergaba por ella. “Me miraba con esos ojos de cachorrito”, recordó más tarde. Los Williams decidieron solucionar el problema de John y, durante un tiempo considerable, un objetivo corporativo no oficial de On-Line Systems fue conseguir que John Harris se acostara con alguien. No fue tan fácil. Aunque las mujeres de su edad podían decir que John Harris era “guapo”, aunque

podía ser muy inteligente verbalmente y sin duda ganaba suficiente dinero para complacer a todos, salvo a las cazafortunas más exigentes, las mujeres no parecían reaccionar sexualmente ante él.

En los alrededores de Oakhurst, por supuesto, incluso encontrar mujeres era un problema. John Harris había aceptado un trabajo a tiempo parcial en el salón de juegos local, pensando que cualquier chica a la que le gustaran los juegos tendría algo en común con él; se propuso quedarse casi todas las horas que el salón de juegos estaba abierto. Pero las chicas que pasaban tiempo en el salón de juegos todavía estaban en el instituto. Cualquier chica local con mucho cerebro se iba a la universidad; las que se quedaban les gustaban los tipos moteros y no se identificaban con los chicos amables que se ponían nerviosos con las mujeres, como John Harris. John invitó a muchas chicas a salir y, por lo general, ellas dijeron que no, probablemente haciéndole sentir como cuando la gente elegía un bando en los partidos de baloncesto y él se quedaba allí sin ser elegido.

Ken prometió cambiar todo eso. “Te voy a conseguir sexo, John Harris”, decía siempre, y aunque John se sentía avergonzado e insistía en que Ken dejara de decir esas cosas, secretamente esperaba que Ken cumpliera su promesa. Pero los contratiempos continuaban.

Cada vez que John salía, había calamidades. Primero, la adolescente que conoció en un restaurante de comida rápida que lo acompañó a comer pizza y no volvió a salir con él. Luego, una mujer que empaquetaba discos para On-Line, una cita organizada por Ken. John se avergonzó a sí mismo al dejar las llaves dentro de su nuevo vehículo con tracción en las cuatro ruedas, tuvo problemas para entrar en el sedán donde iban todos y se mortificó cuando Ken, delante de la mujer, empezó a hacer comentarios groseros sobre lo cachondo que estaba John: “Eso me avergonzó mucho”, dijo John Harris más tarde. Cuando todos volvieron a la casa de los Williams para meterse en el jacuzzi, el 4×4 de John se quedó atascado en la nieve; y finalmente, la chica se encontró con su antiguo novio y se fue con él. Ese fue el final de una cita típica de John Harris.

Ken Williams no se rindió tan fácilmente. Los Williams llevaron a John Harris al Club Med en Haití. ¿Cómo puede un hombre no echar un polvo en el Club Med? Cuando una mujer que no llevaba bikini (se le veían los pechos justo delante) le preguntó a John si le gustaría hacer esnórquel, Ken se rió. ¡Un premio gordo! La mujer tenía unos diez años más, pero tal vez una mujer con experiencia era lo que necesitaba John Harris. El viaje de esnórquel fue muy divertido y, en el camino de vuelta, todas las chicas estaban jugando, poniéndoles las camisetas a los chicos. Roberta agarró el brazo de John y le susurró: "Si no haces algo con esta chica, ¡nunca más te hablaré!".

John Harris reprimió su timidez en ese momento. "Finalmente puse mi brazo alrededor de la chica", recordó más tarde. “Ella dijo: ‘¿Puedo hablar contigo?’ Nos sentamos y ella mencionó nuestra diferencia de edad”. Estaba claro que no había romance en el horizonte. “Había planeado llevarla a navegar, pero después de eso me sentí demasiado avergonzado”, dijo John más tarde.

Ken se volvió aún más atrevido después de Haití. “Hizo bastantes cosas [para encontrarme una mujer]”, dijo John Harris más tarde. En un momento, Williams le preguntó a una camarera en Lake Tahoe: “¿Te gustaría acostarte con un rico veinteañero?”.

Probablemente lo peor de todo sucedió en una despedida de soltero que organizaron para un empleado de On-Line. Ken había contratado a dos strippers. La fiesta se celebró en la oficina y fue indicativa del espíritu desenfadado y de todo vale en la compañía de Ken. La gente bebía mucho; alguien inició un juego en el que intentabas mirar hacia otro lado y arrojar botellas de cerveza a un cubículo lejano. La oficina se cubrió de vidrios rotos y al día siguiente casi todos los que estaban en la fiesta se despertaron con cortes y moretones.

A John le gustó el aspecto de una de las strippers. “Era increíblemente hermosa”, recordó. A John le pareció tímida y le confesó que hasta hace un par de semanas había sido secretaria y que lo hacía porque ganaba muy bien. Bailó alrededor de John Harris, en un momento dado quitándose el sujetador y colocándoselo alrededor de la cabeza.

“Quiero hablar contigo”, dijo Ken, llevándose a John a un lado. “Estoy siendo completamente honesto. Esto es lo que dijo: ‘Es realmente lindo’”.

John solo escuchó.

“Le dije que ganas trescientos mil dólares al año. Ella preguntó si estabas casado”.

Ken no estaba siendo totalmente sincero. Había hecho un trato con la mujer para tener sexo con John Harris. Ken lo arregló todo, diciéndole a John que estaría en el Chez Paree en Fresno, y John se vistió elegante para verla. Ken fue. John y la mujer se retiraron a una mesa del fondo. Ken le dijo a John que los invitaría a tomar algo, pero lo único que ella quería era Seven-Up. Ken le compró a la pareja una botella de Seven-Up. “Las botellas eran caras”, recordó John más tarde. “Veinte dólares la botella”. Fue la primera de muchas botellas de Seven-Ups de veinte dólares. “Esta chica me dejó totalmente fascinado. Era muy fácil hablar con ella. Hablamos de cosas que había hecho antes, de por qué decidió ser stripper. No parecía del tipo de stripper”. Para entonces, Ken ya se había ido y John estaba comprando las botellas de Seven-Up de veinte dólares. El lugar estaba cerrando. Era el momento de la verdad. La chica actuaba como si fuera natural que ella siguiera su camino y que John siguiera el suyo. Así que John se fue a casa. Cuando Ken llamó más tarde y le preguntó si había “ganado”, John recordó más tarde: “No tenía mucho que decir en mi defensa”.

Parecía una situación permanente. Éxito en el Atari, pero sin suerte con las mujeres.

• • • • • • •

A pesar de los problemas femeninos de John Harris, era un nuevo modelo a seguir para una nueva era: la superestrella hacker. Ken se sentaba para entrevistas de revistas y charlaba sobre las virtudes del Atari 800. Los artículos mencionaban a menudo sus ingresos de seis cifras por su acuerdo de regalías del treinta por ciento. Era una posición envidiable, repentinamente moderna. En todo Estados Unidos, jóvenes que se describían a sí mismos como hackers trabajaban en sus obras maestras: era el equivalente de la nueva era a todos aquellos jóvenes de los años cuarenta que intentaban escribir la Gran Novela Americana. Las posibilidades de que un juego superventas pudiera llegar a través del espejo de On-Line, aunque no eran grandes, eran algo mejores que las de una novela superventas no solicitada.

Ken se dio cuenta de que estaba compitiendo con otras compañías de la Hermandad por estos programadores. A medida que más gente aprendía la magia del lenguaje ensamblador de Apple y Atari que era única cuando Ken Williams comenzó, el consumidor de computadoras domésticas se volvía más exigente con lo que compraba. Otras compañías además de On-Line estaban publicando aventuras gráficas, habiendo descubierto sus propios trucos para poner docenas de imágenes con texto en los discos de Apple. Además, una nueva empresa de Cambridge llamada Infocom, que utilizaba sólo texto, había desarrollado un intérprete avanzado que aceptaba vocabularios extensos de palabras, en oraciones completas. La empresa fue fundada por hackers del MIT. Su primer juego de microcomputadoras, copiado directamente del juego que habían escrito para divertirse en una de las computadoras de Tech Square, fue Zork, una versión supercargada del cuento de mazmorras original de Adventure escrito por Crowther y Woods en Stanford. Se vendía de forma demencial.

Era un indicador de lo rápido que se movía el mercado de los juegos de computadora. Lo que era brillante un año parecía anticuado al siguiente. Los hackers de Apple y Atari habían llevado a las máquinas mucho más allá de sus límites. Por ejemplo, sólo habían pasado unos meses desde su introducción que el programa Skeet Shoot de On-Line parecía tan tosco que resultaba embarazoso, y Ken lo eliminó de la línea de productos. Threshold, por ejemplo, hizo volar por los aires ese estándar anterior. Y un hacker llamado Bill Budge escribió un programa que simulaba una máquina de pinball, Raster Blaster, que superaba prácticamente todo lo que On-Line tenía para ofrecer en Apple.

Ken Williams sabía que On-Line tenía que presentarse como un lugar deseable para trabajar. Él y su personal armaron un paquete impreso lleno de promesas y sueños para las futuras superestrellas del software. Curiosamente, los atractivos que ofrecía On-Line tenían poco que ver con la ética hacker. El paquete no enfatizaba la feliz comunidad del campamento de verano alrededor de On-Line. En cambio, parecía casi un himno a Mammon.

Una parte del paquete se titulaba “Preguntas y respuestas”.

> PREGUNTA: ¿Por qué debería publicar con On-Line (y no con otra persona)?
>
> RESPUESTA: Una muy buena razón es el dinero. ON-LINE paga las regalías más altas y regulares del sector... ¡Nuestro trabajo es hacerte la vida más fácil!
>
> PREGUNTA: ¿Por qué no publicar yo mismo?
>
> RESPUESTA: Con ON-LINE su producto recibirá soporte de un personal técnico altamente capacitado. Esto le deja tiempo libre para cosas más importantes como cruceros por el Caribe, esquiar en Aspen y todas las demás actividades "duras" de la vida. En pocas palabras, nosotros hacemos todo el trabajo... Lo único que le pedimos es que esté disponible para nosotros en caso de que ocurra algún error. Aparte de eso, ¡simplemente siéntese y observe cómo llega el dinero!

También en el paquete había una carta de Ken Williams ("Presidente de la Junta Directiva") explicando por qué On-Line Systems era la empresa de marketing más profesional y eficaz. Ken se mostró muy satisfecho con la experiencia técnica de su equipo, y citó al personal de programación de Schwader, Davis y Stephenson. También había una carta del director de ventas de On-Line: “Somos los mejores y queremos que sólo los mejores estén en nuestro equipo. Si encajas en esta sencilla descripción, ven a respirar el aire enrarecido con nosotros en la cima. El éxito es embriagador. ¿Lo puedes soportar?”. Una nota del Departamento de Adquisiciones de Software resumía el mensaje a los posibles programadores: “Nos interesan porque eres la savia de nuestro negocio. La programación se ha convertido en un producto de primera calidad”.

Fue una transformación bastante grande con respecto a los días en que un hacker estaría más que satisfecho de ver a alguien apreciar el arte de su software. Ahora que había un mercado, el mundo real había cambiado el hackerismo. Tal vez fuera una compensación necesaria por los beneficios de la disponibilidad generalizada de computadoras. Observa todas las maravillosas transformaciones que las computadoras habían realizado en las vidas de las personas de la comunidad de On-Line.

Ken estaba enormemente orgulloso de estas transformaciones. Parecían cumplir la brillante promesa del sueño hacker. No sólo estaba prosperando, sino que él y las otras compañías de la Hermandad lo estaban haciendo con una mentalidad desinteresada y de nueva era... ¡eran los pioneros de la Nueva

América! Y lo que era más, a medida que pasaban los meses se hacía cada vez más evidente que los ordenadores eran una industria en auge como nadie había visto desde la industria del automóvil. Todo el mundo quería una parte de ella. Apple Computer, que parecía una empresa cuestionable cuando Ken vio por primera vez el Apple II, estaba en camino de convertirse en una empresa de la lista Fortune 500, más rápidamente que cualquier otra empresa en la historia. Los capitalistas de riesgo se estaban centrando en el campo de la informática y parecían identificar el software (cosas para hacer funcionar estos ordenadores) como la inversión especulativa más atractiva del país. Dado que los juegos eran, por el volumen de disquetes vendidos, las aplicaciones informáticas más vendidas, y las empresas de la Hermandad tenían entre ellas un porcentaje considerable del mercado de los juegos de ordenador, las ofertas de inversión y las adquisiciones llegaban con tanta frecuencia como los paquetes de juegos nuevos. Aunque a Ken le encantaba hablar con estos pretendientes adinerados, cuyos nombres aparecían a menudo en The Wall Street Journal, se aferró a su empresa. Los teléfonos de la Hermandad sonaban a menudo con el último informe de una oferta de compra: “¡Dijo que pagaría diez millones!”. “¡Bueno, me acaban de ofrecer diez millones por la mitad de la empresa!”. “¡Ah, y rechacé a fulano por esa cantidad!”. Ken se reunía con estos pretendientes en reuniones de desayuno en el aeropuerto, pero los respectivos ejecutivos se iban en avión a sus destinos finales sin acuerdos de compra. Ken Williams se divertía demasiado cambiando la vida de las personas y conduciendo al trabajo en su nuevo Porsche 928 rojo fuego como para plantearse renunciar a él.

Capítulo 18. Frogger

A medida que avanzaba 1982 y se acercaba el segundo aniversario de su empresa, Ken Williams empezaba a perder la paciencia con John Harris y con los hackers jóvenes en general. Ya no tenía tiempo ni ganas de dedicar horas de asistencia técnica a sus hackers. Empezó a considerar las preguntas que le hacían sus programadores (¿Cómo puedo poner esto en la pantalla sin que parpadee? ¿Cómo puedo desplazar objetos horizontalmente? ¿Cómo me deshago de este error?) como distracciones de lo que se estaba convirtiendo en su actividad principal: hackear sistemas en línea a medida que crecían a pasos agigantados. Hasta ahora, cuando un programador llamaba a Ken y aullaba frenéticamente que estaba atascado en alguna subrutina, Ken se acercaba, lloraba con él y trasteaba con el programa, haciendo lo que fuera necesario para hacer feliz a su hacker. Esos días estaban terminando.

Ken no veía que el cambio de actitud hiciera que su empresa fuera menos idealista. Ken Williams seguía creyendo que On-Line estaba cambiando vidas a través de la computadora, tanto las vidas de sus trabajadores como las de sus clientes. Era el comienzo de un milenio informático. Pero Ken Williams no estaba seguro de que el hacker fuera la figura central de esta era dorada. Especialmente un hacker como John Harris.

La separación entre Ken Williams y John Harris simbolizó algo que estaba ocurriendo en toda la industria del software para computadoras domésticas. Al principio, los objetivos artísticos del hacker coincidían perfectamente con el mercado, porque el mercado no tenía expectativas y los hackers podían crear alegremente los juegos que querían jugar y adornar los programas comerciales con las ingeniosas características que mostraban su arte.

Pero a medida que más personas no técnicas compraban computadoras, las cosas que impresionaban a los hackers no eran tan esenciales. Si bien los programas en sí tenían que mantener un cierto estándar de calidad, era muy posible que los estándares más exigentes (los aplicados por un hacker que quería agregar una característica más o no abandonaba un proyecto hasta que fuera demostrablemente más rápido que cualquier otra cosa) fueran probablemente contraproducentes. Lo que parecía más importante era el marketing. Había muchos programas brillantes que nadie conocía. A veces los hackers escribían programas y los ponían en el dominio público, los regalaban con la misma facilidad con la que John Harris había prestado su primer ejemplar de Jawbreaker a los chicos de la tienda de informática de Fresno. Pero rara vez la gente pedía programas de dominio público por su nombre: querían los que veían anunciados y comentados en revistas, o demostraciones en las tiendas de informática. No era tan importante tener algoritmos increíblemente inteligentes. Los usuarios se conformaban con los más comunes.

La ética hacker, por supuesto, sostenía que cada programa debía ser tan bueno como se pudiera hacer (o mejor), infinitamente flexible, admirado por su brillantez de concepto y ejecución, y diseñado para ampliar los poderes del usuario. Vender programas informáticos como si fueran pasta de dientes era una herejía. Pero se estaba haciendo. Considere la receta para el éxito ofrecida por uno de los miembros de un panel de capitalistas de riesgo de alta tecnología, reunidos en una feria de software en 1982: "Puedo resumir lo que se necesita en tres palabras: marketing, marketing, marketing". Cuando los ordenadores se vendan como tostadoras, los programas se venderán como pasta de dientes. A pesar de la ética hacker.

Ken Williams anhelaba los best-sellers, juegos cuyos nombres tuvieran el impacto de las marcas. Así que cuando su programador estrella, John Harris, mencionó que le gustaría probar a convertir

un popular juego de arcade llamado Frogger en el Atari Home Computer, a Ken le gustó la idea. Frogger era un juego simple pero fascinante en el que el jugador intentaba manipular a una simpática ranita por una carretera muy transitada y un arroyo haciéndola saltar sobre troncos y tortugas; el juego era popular y, si se hackeaba bien, bien podría convertirse en un juego de ordenador de gran éxito. “John Harris lo vio y dijo que era realmente genial. Me dijo que podía programarlo en una semana. Acepté; parecía trivial”, recordó Ken más tarde.

En lugar de que Harris copiara el programa y le diera otro nombre, Ken Williams siguió las reglas de la corporación. Llamó al propietario de los derechos del juego, la división Sega del conglomerado Gulf & Western. Sega no parecía entender el valor de su propiedad, y Ken se las arregló para adquirir los derechos de los discos y casetes de ordenador por una mísera cuota de regalías del diez por ciento. (Sega cedió los derechos de los cartuchos a la empresa de juegos Parker Brothers; los comercializadores del Monopoly estaban entrando en el mercado de los videojuegos). Puso a John Harris a trabajar inmediatamente en la conversión del juego al ordenador Atari. También asignó a un programador para que hiciera una versión para Apple, pero como los gráficos de Apple no se adaptaban bien al juego, sería el Atari el que mostraría la excelencia de la empresa de Ken.

John Harris supuso que sería un proyecto rápido y sucio de tres semanas (su alarde original de una semana había sido inútil) hacer una versión Atari perfectamente admirable de Frogger. Este era el tipo de ilusión con la que los hackers suelen empezar los proyectos. Trabajando en la oficina que había instalado en el más pequeño de los tres dormitorios de su destartalada casa de madera anaranjada (una habitación abarrotada de papeles, hardware desechado y bolsas de papas fritas), John puso los gráficos en la pantalla en poco tiempo; durante ese período, recordó más tarde, "me pegué las manos al teclado. Una vez empecé a programar a las tres de la tarde. Después de escribir código durante un rato, miré hacia fuera y todavía había luz afuera y pensé: "Parece que he estado escribiendo durante más de unas horas". Y, por supuesto, ya había pasado toda la noche y eso había sido a la mañana siguiente".

El trabajo avanzaba rápidamente y el programa estaba tomando forma maravillosamente. Un amigo de John en San Diego había escrito algunas rutinas para generar música continua, utilizando el chip sintetizador de sonido de tres voces del Atari para mezclar las notas del tema original de Frogger con Camptown Races, todo con el alegre contrapunto de un calíope. Las formas gráficas de Harris nunca habían sido mejores: la rana saltando, los pequeños coches deportivos y camiones en la carretera, las tortugas que se zambullen y los caimanes de aspecto ridículo en el agua... cada detalle definido con cariño en tablas de formas, trabajado en subrutinas de lenguaje ensamblador e integrado con destreza en el juego. Era el tipo de juego, creía Harris, que sólo una persona enamorada de los juegos podría implementar. Nadie, excepto un verdadero hacker, lo abordaría con la intensidad lunática y la meticulosa exactitud artística de John Harris.

No resultó ser un proyecto rápido y sucio de tres semanas, pero nadie lo hubiese esperado así. El software siempre lleva más tiempo del que uno espera. Sin embargo, casi dos meses después de empezar el proyecto, John ya había superado el obstáculo. Decidió faltar al trabajo durante un par de días para volver a San Diego para la Software Expo, un evento benéfico para la distrofia muscular. Como artista de software líder, John iba a exhibir su trabajo, incluido el Frogger casi terminado. Así que John Harris metió el Frogger de prelanzamiento en su colección de software y se llevó la caja entera al sur de California.

Cuando se viaja con un cargamento tan valioso como ese, se requería un cuidado extremo. Además de incluir la única versión de Frogger, el programa más importante que John Harris había escrito jamás (John tenía una copia de seguridad, por supuesto, pero la llevaba consigo por si el disco principal no arrancaba), la biblioteca de John incluía casi todos los discos que poseía, discos cargados con utilidades de software: ensambladores modificados por él mismo, rutinas para modificar archivos, generadores de música, rutinas de animación, tablas de formas... una vida joven de herramientas, el equivalente para él de todo el cajón de programas en cinta de papel para el PDP-1 del MIT. Uno no podía darle la espalda a una colección tan inestimable como esa; uno la tenía en la mano casi en todo momento. De lo contrario, en el momento en que uno se olvida de tenerlo en la mano y le da la espalda (por ejemplo, durante un momento de conversación embelesada con un admirador), como dice la Ley de Murphy (“Todo lo que puede salir mal, saldrá mal”), la valiosa biblioteca de software de uno podría desaparecer trágicamente.

Eso fue precisamente lo que le pasó a John Harris en la Software Expo.

En el instante en que John Harris terminó su interesante conversación y vio que su colección de software había desaparecido, supo que su alma había sido herida. Nada era más importante para John que los disquetes en esa caja, y sintió profundamente el vacío. No era como si la computadora hubiera devorado un disquete y él pudiera entrar en modo maratón durante unos días para restaurar lo que había perdido en la pantalla. Esta era una obra maestra en toda regla totalmente borrada. Y aún peor, las herramientas con las que había creado la obra maestra también habían desaparecido. No había peor desastre imaginable.

John Harris cayó en una profunda depresión.

Estaba demasiado molesto para encender su Atari y comenzar la laboriosa tarea de reescribir Frogger cuando regresó a Oakhurst. Durante los dos meses siguientes, no escribió más que diez líneas de código fuente. Era difícil incluso sentarse frente a la computadora. Pasaba casi todo el día, todos los días, en la única sala de juegos de Oakhurst, una pequeña tienda en un minúsculo centro comercial al otro lado de la calle del edificio de oficinas de dos pisos al que se estaba mudando On-Line. En comparación con las salas de juegos, esta era un agujero, con paredes oscuras y nada para decorar excepto las propias máquinas de videojuegos; y ni siquiera los últimos modelos. Pero era el hogar de John. Consiguió un trabajo a tiempo parcial como cajero. Cambiaba fichas de juego por monedas de veinticinco centavos, y cuando no estaba de servicio jugaba a Starpath y Robotron y Berzerk y Tempest. Parecía que eso ayudaba. Otras veces se subía a su camioneta 4x4, salía de la carretera, buscaba la colina más grande que pudiera encontrar e intentaba conducir hasta la cima. De hecho, hacía cualquier cosa, menos programar.

“Pasaba casi todas las horas de cada día en el salón de juegos esperando a que entrara alguna chica”, recordó más tarde. “Volvía a casa y jugaba a un juego en mi computadora y luego intentaba introducir el disco del programa y comenzar a programar como si estuviera jugando el juego”. Nada de eso funcionó. “No pude motivarme para escribir dos líneas de código fuente”.

El corazón de Ken Williams no se conmovió ante la pérdida de John Harris. A Ken le resultaba difícil sentir simpatía por un muchacho de veinte años al que le pagaba varios miles de dólares al mes en regalías. Ken sentía un sentimiento de amistad hacia John, pero también había desarrollado una teoría sobre los amigos y los negocios. “Todo es personal y buenos amigos hasta unos diez mil dólares”, explicó Ken más tarde. “Una vez que se superan los diez mil dólares, la amistad no importa”. Las posibles ganancias de Frogger valían muchas veces ese umbral de cinco cifras.

Incluso antes de que John le hubiera demostrado una vez más su idiotez a Ken Williams con su descuido en la Software Expo, Ken había estado impaciente con su programador estrella. Ken pensaba que John debería haber escrito Frogger en menos de un mes para empezar. “John Harris es un perfeccionista”, dijo Ken Williams más tarde. “Un hacker. Seguirá trabajando en un proyecto durante dos meses después de que cualquier otro hubiera parado. Le gusta la satisfacción del ego de tener algo mejor que cualquier otra cosa en el mercado”. Ya era bastante malo, pero el hecho de que John no estuviera trabajando en absoluto ahora, sólo porque había sufrido un revés, enloquecía a Ken. “Decía que no estaba entusiasmado”, recuerda Ken. “¡Luego lo encontraba en la sala de juegos, trabajando por fichas!”.

Delante de los amigos de John, Ken hacía comentarios desagradables sobre lo que se estaba demorando Frogger. Ken ponía a John demasiado nervioso como para pensar en réplicas concisas en el momento. Sólo lejos de Ken podía John Harris darse cuenta de que debería haber dicho que no era su empleado, que era un programador autónomo. No le había garantizado a Ken ninguna fecha de entrega. John podía hacer lo que quisiera. Eso era lo que debería haber dicho. En cambio, John Harris se sintió mal.

Fue una tortura, pero finalmente John se arrastró hasta el Atari y comenzó a reescribir el programa. Finalmente, recreó su trabajo anterior, con algunos adornos adicionales también. Cuarenta y cuatro colores, las rutinas gráficas del misil jugador totalmente redefinidas y un par de trucos ingeniosos que lograron que los ocho bits del chip Atari 6502 emularan diez bits. El amigo de John en San Diego incluso había mejorado un poco la banda sonora con tres voces simultáneas. En general, la versión de John Harris se veía incluso mejor que el juego de arcade, una hazaña asombrosa ya que los juegos de arcade usaban chips diseñados a medida para gráficar a alta velocidad con colores sólidos, y casi nunca eran igualados por los ordenadores domésticos menos potentes (aunque más versátiles). Incluso programadores experimentados como Jeff Stephenson estaban impresionados.

El período oscuro había terminado, pero algo había cambiado en la relación entre Ken y John. Era emblemático de la forma en que On-Line estaba cambiando, convirtiéndose más en una burocracia que en un campamento de verano para hackers. Mientras que el procedimiento para lanzar los juegos anteriores de John había sido una prueba improvisada en el lugar ("¡Eh! ¡Tenemos un juego para probar hoy! Si a todos les gusta, lo lanzamos"), ahora Ken tenía un departamento separado para probar los juegos antes del lanzamiento. A John le parecía que hacían falta unos cincuenta intercambios de notas internas antes de que alguien se animara a decir que le gustaba un juego. También había atascos con el empaquetado, la comercialización y la protección de copias. Nadie sabía exactamente cómo, pero pasaron más de dos meses (dos meses después de que John hubiera entregado su Frogger totalmente terminado) para que el juego saliera al mercado.

Cuando finalmente salió al mercado, todo el mundo reconoció que Frogger era una conversión estupenda de una máquina recreativa a un ordenador doméstico. El cheque de John por las regalías del primer mes fue de treinta y seis mil dólares, y el programa llegó al número uno de la nueva «Lista de los mejores» de programas de Softsel Distributors (que se compilaba semanalmente y seguía el modelo de la lista de récords de Billboard), y permaneció allí durante meses.

Sin embargo, Ken Williams nunca olvidó los problemas que John Harris le había causado durante la etapa de depresión, cuando parecía que John nunca entregaría un Frogger que funcionara. Y en el verano de 1982, Ken empezó a hacer planes para el día en que se vería libre de todos los John

Harris del mundo. En lo que respecta a Ken Williams, la era de los hackers había terminado. Y su fin había llegado justo a tiempo.

• • • • • • • •

Al igual que su primer modelo a seguir, Jonas Cord de The Carpetbaggers, a Ken Williams le encantaba hacer tratos. Llamaba por teléfono a un posible programador y le decía, sin vergüenza alguna y con un ligero sentido de parodia, "¿Por qué no me dejas hacerte rico?". También le gustaba tratar con ejecutivos de corporaciones gigantes en igualdad de condiciones. En 1982, uno de los primeros años de auge de la revolución informática, Ken Williams habló con mucha gente, y los tipos de tratos que hizo indicaban en qué tipo de negocio se estaba convirtiendo el software para ordenadores domésticos y qué lugar, si es que había alguno, tendrían los hackers o la ética hacker en el negocio.

"On-Line es una locura", dijo Williams ese verano. “Tengo esta filosofía de que o quiero fingir ser IBM o no estar aquí”.

Soñaba con tener un impacto nacional en el mercado masivo. En el verano de 1982, eso significaba que la máquina Atari VCS – la máquina de juegos dedicada para la cual los juegos superventas no debían contarse en decenas de miles, como el software de Apple – sino en millones.

Atari consideraba que el funcionamiento de su máquina VCS era un secreto guardado con más celo que la fórmula de la Coca-Cola. Si hubiera sido la fórmula de un refresco, el plan esquemático de la VCS (qué ubicación de memoria en el chip activaba el color en la pantalla y qué punto de acceso activaba el sonido) bien podría haber permanecido en las bóvedas de Atari. Pero esto era la industria informática, donde descifrar códigos había sido un pasatiempo desde los días de violar cerraduras en el MIT. Con el incentivo adicional de ganancias vertiginosas que podría obtener cualquiera capaz de superar las más bien mundanas ofertas de software que la complaciente Atari vendía para su máquina, era sólo cuestión de tiempo antes de que alguien descifrara los secretos de la VCS (al igual que los secretos del Atari 800).

Las primeras empresas que se desafiaron a Atari con su VCS fueron, de hecho, empresas de nueva creación formadas por antiguos programadores de Atari a los que su presidente había llamado “diseñadores de toallas”. Casi todos los genios del VCS de Atari abandonaron el barco a principios de los años 80. No fue una pérdida pequeña, porque la máquina VCS tenía una memoria desesperadamente limitada y escribir juegos en ella requería habilidades tan perfeccionadas como las que se requieren para la composición de haikus. Sin embargo, los programadores de Atari que se marcharon sabían cómo ampliar la máquina mucho más allá de sus limitaciones; los juegos que escribieron para sus propias empresas hicieron que Atari pareciera ridícula. La calidad mejorada de los juegos prolongó la vida comercial del VCS durante años. Fue una justificación sorprendente de la insistencia de los hackers en que cuando a que los manuales y otros “secretos” se difundan libremente: los creadores se divierten más, el desafío es mayor, la industria se beneficia y los usuarios se ven recompensados con productos muy superiores.

Mientras tanto, otras compañías estaban “realizando ingeniería inversa” del VCS, diseccionándolo con osciloscopios y dispositivos de tecnología indescriptiblemente alta hasta que entendieron sus secretos. Una de esas compañías fue Tiger Toys, una compañía con sede en Chicago que contactó a Ken Williams para establecer un acuerdo para compartir su talento como programador.

Williams envió a tres hackers a Chicago, donde Tiger Toys les enseñó lo difícil que era programar el VCS. Había que ser tacaño con el código, había que contar ciclos de la máquina para espaciar los movimientos de las cosas. John Harris en particular lo odiaba, a pesar de que él y Roberta Williams se habían sentado una noche y habían ideado un nuevo y elegante diseño de VCS para Jawbreaker que se parecía menos a Pac-Man. John Harris estaba acostumbrado a las rutinas mucho más rápidas del ordenador Atari 800, y estaba indignado de que esta otra máquina se negara a aceptar rutinas similares.

Consideraba que el VCS era ridículo. Pero John realmente quería hacer un programa que dejara a la versión VCS de Pac-Man por las nubes, y con el nuevo esquema de Jawbreaker, en su opinión, pudo lograr esa tarea. El Pac-Man VCS de Atari estaba lleno de parpadeos, un gran perdedor; el programa VCS de John no tenía parpadeos, era colorido y era increíblemente rápido.

Los negocios de Ken Williams no se detuvieron en el mercado de VCS. Como los juegos de computadora estaban teniendo tanto éxito como las películas, pudo establecer vínculos con esa industria. El mundialmente famoso creador de los Muppets, Jim Henson, iba a estrenar esa Navidad una película de 20 millones de dólares llamada Dark Crystal que tenía visos de ser un éxito de taquilla. Ken y Henson llegaron a un acuerdo.

Mientras que Ken supuso que la idea de vincular un juego de computadora a una película inédita era arriesgada (¿y si la película fracasaba?), a Roberta Williams le encantó la idea de escribir un juego de aventuras basado en los personajes de Dark Crystal. Consideraba que los juegos de ordenador eran una faceta del mundo del entretenimiento tanto como las películas y la televisión, y pensó que era natural que su género se fusionara con esas glamorosas contrapartes. De hecho, otras empresas de videojuegos y ordenadores estaban trabajando en proyectos relacionados con películas. Estaba E.T. de Atari, M.A.S.H. de Fox Videogames y The Empire Strikes Back de Parker Brothers. Una empresa de juegos de ordenador llamada DataSoft incluso estaba trabajando en un juego de aventuras basado en el programa de televisión Dallas. Esto suponía un gran paso respecto de los primeros tiempos, cuando todo lo que un programador tenía para trabajar era la creatividad. Ahora podía trabajar con una propiedad rentable.

Si Dark Crystal no era precisamente una gran liga, el siguiente acuerdo sí lo era. Para éste, Ken Williams estaba tratando con la empresa más grande de todas.

IBM.

International Business Machines, de tú a tú con la empresa de Coarsegold, California, que no existía hace dos años. Los empleados de IBM, vestidos con camisas blancas y corbatas oscuras, que trabajaban en lotes, llegaban a la nueva sede corporativa de Ken, que consistía en una serie de oficinas en el mismo edificio que albergaba la pequeña oficina donde los habitantes de Oakhurst y Coarsegold pagaban sus facturas de electricidad, una pequeña tienda de muebles en la planta baja y un salón de belleza junto a la oficina donde Ken se encargaba del marketing y la publicidad.

Para los trabajadores de On-Line, los hackers y los nativos de Oakhurst vestidos con pantalones cortos y camisetas de los campamentos de verano, el comportamiento encubierto de IBM era absurdo. Todo era solemnemente supersecreto. Antes de que IBM divulgara siquiera un atisbo de sus intenciones, su personal, con cara de póquer, insistió en que todos los que pudieran saber algo sobre el acuerdo (y esto debía limitarse al menor número posible de personas) firmaran extensos y vinculantes formularios de confidencialidad que preveían severas torturas y lobotomías frontales completas, casi, a cualquiera que filtrara el nombre de la empresa de las tres iniciales o sus planes.

Las predicciones del autor de Computer Lib Ted Nelson y otros de que la revolución de la computadora personal pondría a IBM "en desorden" habían demostrado ser una subestimación patética de la empresa monolítica. El enorme gigante de las empresas de computadoras había demostrado ser más ágil de lo que nadie había esperado. En 1981, había anunciado su propia computadora, la IBM "PC", y el mero espectro de esta entrada llevó a muchos en la pequeña industria de las computadoras a hacer preparativos para darse por vencidos y morir por IBM, lo que hicieron rápidamente cuando la máquina PC de IBM se puso en el mercado. Incluso la gente que odiaba a IBM y su filosofía de procesamiento por lotes se dio por vencida y cayó muerta, porque IBM había hecho algo que representaba un cambio total respecto de todo lo que había defendido anteriormente: abrió sus máquinas. Alentó a personas ajenas a escribir software. Incluso contrató a empresas externas para que ayudaran a diseñarlo, empresas como Microsoft, encabezada por Bill Gates (el autor de la carta original sobre piratería de software, dirigida a los copiadores de Homebrew Altair BASIC). Gates escribió el sistema operativo de IBM que casi instantáneamente se convirtió en un nuevo estándar de la industria. Era casi como si IBM hubiera estudiado la Ética Hacker y hubiera decidido que, en este caso, era un buen sentido comercial aplicarla.

Sin embargo, IBM no planeaba aplicar demasiado la Ética Hacker. Todavía valoraba el secreto como forma de vida. Así que IBM esperó hasta que todos los formularios de confidencialidad estuvieran firmados antes de que sus hombres de camisa blanca le dijeran a Ken Williams lo que tenían en mente. IBM estaba planeando una nueva máquina para el hogar, más barata y mejor para jugar que el PC. El nombre en código era Peanut, pero con el tiempo se conocería como PCjr. ¿Le gustaría a On-Line crear un nuevo tipo de intérprete de aventuras, más sofisticado que todo lo que había existido hasta entonces? Y también escribir un programa de procesamiento de textos fácil de usar para el PCjr, pensó Ken, que podrían hacerlo sin problema, y mientras Roberta comenzaba a trazar otra trama de aventuras, Ken se dedicó a contratar a un equipo ultrasecreto de magos para que hackearan el código del proyecto.

A On-Line le costaría mucho dinero participar en algunas de estas empresas de alto riesgo. Pero Ken Williams se había ocupado de eso con el acuerdo más importante de todos. Capital de riesgo. "Nunca había oído hablar del capital de riesgo", dijo Ken Williams más tarde. "Tuve que convencerme de aceptarlo". Aun así, On-Line estaba gastando dinero muy rápidamente, y los 1,2 millones de dólares que la empresa recibió de una firma de Boston llamada TA Associates (más doscientos mil dólares para Ken y Roberta personalmente) fueron esenciales para mantener el flujo de efectivo. A cambio, TA obtuvo el veinticuatro por ciento de la empresa y derechos de consulta sobre varios aspectos del negocio.

La mujer de TA que hizo el trato era Jacky Morby, una mujer vibrante y canosa con rasgos precisos, una intensidad estudiada y la capacidad de insinuarse como una madrina lejana de la empresa. Jacky Morby tenía mucha experiencia en situaciones en las que empresarios brillantes crean empresas que crecen tan rápido que amenazan con salirse de control, y de inmediato le aconsejó a Ken Williams, de tal manera que él supo que no era un simple consejo casual, que buscara una gestión profesional. Reconoció que Ken no era un tipo con MBA, no alguien que pudiera cuidar adecuadamente su empresa para que ocupara su lugar en la línea tradicional de empresas que hacen que este país sea grande y que las firmas de capital de riesgo como TA se hagan muy ricas. Si On-Line Systems saliera a bolsa y pusiera a todos en modo Creso, tendría que haber un timonel firme que la guiara en las aguas que se avecinaban. El timón de Ken estaba torcido. No dejaba de virar hacia planes

descabellados, negocios disparatados y fracasos en campamentos de verano de hackers. Alguien tendría que venir y proporcionar un nuevo timón.

La idea no le resultó desagradable a Ken, que ya en marzo de 1981 había anunciado a Softalk que se "despedía del personal de On-Line con la esperanza de poder hacer algo de programación". Y sin duda estaba claro que había que hacer algo con el lío de gestión que se estaba espesando a medida que la empresa vendía más software, cerraba más contratos, intentaba conseguir más programadores y manejaba más papeles, aunque gran parte de ellos estuvieran en forma de datos manejados en ordenadores Apple.

El problema surgió porque Ken hackeaba On-Line como si fuera un sistema informático, modificando un plan de marketing aquí, depurando la contabilidad allá. Al igual que su hacking de computadoras, que se caracterizaba por ráfagas explosivas de innovación y falta de atención a los detalles, su estilo empresarial estaba marcado por destellos de perspicacia y fallas a la hora de llevar a cabo ideas. Fue uno de los primeros en reconocer el valor de un procesador de textos de bajo coste para la Apple (una culminación de la idea que los hackers del Club de Ferrocarriles a Escala del MIT tuvieron cuando escribieron "Máquina de escribir costosa" en el TX-0), y tuvo la paciencia de apoyar el programa a través de innumerables revisiones; el programa, finalmente llamado "Screenwriter II", recaudaría más de un millón de dólares en ventas. Pero sus amigables competidores se reirían de su hábito de escribir enormes cheques de regalías para programadores en la misma chequera que usaba para sus cuentas de supermercado. Ayudaría a desarrollar un programa llamado "The Dictionary", que corrige la ortografía de un usuario de Apple, pero luego colocaría un anuncio de revista para el producto que contenía diez errores ortográficos, incluida una falta de ortografía de la palabra "misspell".

La nueva oficina de Ken estaba prácticamente sepultada en basura. Un nuevo empleado informó más tarde que al ver la habitación por primera vez, supuso que alguien se había olvidado de sacar una enorme y sucia pila de basura. Luego vio a Ken trabajando y lo entendió. El ejecutivo de veintiocho años, con su habitual camiseta azul descolorida de Apple Computer y unos vaqueros desgastados con un agujero en la rodilla, se sentaba detrás del escritorio y mantenía una conversación con empleados o gente por teléfono mientras revisaba papeles. La camiseta se le ponía por encima de la protuberante barriga de Ken, que estaba experimentando un crecimiento casi tan espectacular como las cifras de ventas de su empresa. Avanzando a paso de vértigo, echaba un vistazo a los contratos importantes y los tiraba al montón con indiferencia. Autores y proveedores hablaban por teléfono constantemente, preguntándose qué había pasado con sus contratos. En On-Line había grandes proyectos en marcha para los que no se había firmado ningún contrato. Nadie parecía saber qué programador estaba haciendo qué; en un caso, dos programadores en diferentes partes del país estaban trabajando en conversiones de juegos idénticos. Los discos maestros, algunos sin copias de seguridad, algunos de ellos discos de IBM de alto secreto, estaban apilados en el suelo de la casa de Ken, donde uno de sus hijos podría recogerlos o su perro orinar sobre ellos. No, Ken Williams no era una persona detallista.

Él también lo sabía. Ken Williams llegó a la conclusión de que su empresa había crecido tanto que tenía que ser dirigida de una manera más tradicional por alguien sin tendencias hacker. Finalmente, se le ocurrió un candidato: su antiguo jefe, Dick Sunderland.

Ken conocía a Dick Sunderland como representante de las vagas cualidades que debería tener una empresa respetable, cualidades de las que On-Line carecía notoriamente: previsibilidad, orden,

control, planificación cuidadosa, perspectiva uniforme, decoro, adhesión a las directrices y una jerarquía estructurada. No era casualidad que estas cualidades faltantes fueran cosas que los hackers detestaban. Si Ken se hubiera propuesto encontrar a alguien que representara mejor la antítesis de la ética hacker, habría tenido dificultades para superar a su antiguo jefe. El acto era similar a que alguien reconociera que estaba enfermo y eligiera perversamente la medicina de peor sabor como reconstituyente.

Había algo más insidioso en la elección también. Una de las razones por las que Ken había dejado Informatics varios años antes era que Dick Sunderland le había dicho: “Ken, no tienes potencial de gestión”. Por lo tanto, la idea de ser el jefe de Dick Sunderland atraía mucho la inclinación de Ken por derribar el orden establecido.

A Dick Sunderland, la perspectiva de trabajar para Ken Williams le pareció inicialmente absurda. “¡Ven y dirige mi empresa!”, le había dicho Ken por teléfono desde este complejo montañoso cerca de Yosemite. Esa no era forma de reclutar ejecutivos, pensó Dick. No hay manera, se dijo, de que me involucre en un trato como éste. Dick estaba terminando un programa de MBA, un cambio que creía que lo pondría en la mira de los puestos más altos en Informatics. Pero cuando Ken lo llamó por segunda vez, Sunderland ya estaba preocupado por su futuro en Informatics y había estado pensando en el floreciente campo de la microcomputadora. A principios de junio, Dick llegó en coche y almorzó en Broken Bit con el variopinto grupo de recauchutados de Oakhurst y desertores universitarios que componían la alta dirección de Ken. Observó el acuerdo de capital de riesgo y quedó impresionado. Al final, llegó a pensar que On-Line, como dijo más tarde, "tenía un potencial increíble, algo con lo que podía trabajar. Podía aportar lo que faltaba: un liderazgo cohesivo para que las cosas funcionaran". Dick se dio cuenta de que la industria del software doméstico era "nueva, como la arcilla... se podía moldear y hacer que funcionara, crear un éxito... ¡BOOM! Era la oportunidad de mi vida".

Por otro lado, iba a trabajar para Ken Williams. Durante más de un mes, Dick y su esposa April pasarían horas sentados en el patio trasero de la casa de Los Ángeles que habían decorado cuidadosamente a lo largo de los años, dándole vueltas a esta fantasía que obligaría a evacuarlos de la casa, y quedaría claro que el riesgo número uno era la personalidad de este programador salvaje convertido en zar del software. Dick consultó a profesionales para analizar cómo sería tener un gerente cuidadoso trabajando para este empresario imprudente; habló con expertos en gestión. incluso un psiquiatra. Sunderland se convenció de que podía manejar el Problema Ken.

El 1 de septiembre de 1982, Dick Sunderland comenzó como presidente de On-Line Systems, que coincidentemente también estaba cambiando su nombre. Reflejando la proximidad de Yosemite, la compañía ahora se llamaría Sierra On-Line, y el nuevo logotipo tenía un dibujo de Half-Dome Mountain en un círculo. Un cambio para adaptarse a la nueva era.

Una semana antes de que Dick llegara, Ken se sentía expansivo. Fue el día en que se acercó para dar su bendición al hacker que había "audicionado" con su juego Wall Wars. Después de ese encuentro, habló con un visitante sobre el potencial estrellato de sus pupilos. Admitió que algunos de sus autores se habían convertido en marcas, casi como estrellas de rock. "Si lanzo un juego y le pongo el nombre de John Harris, venderá un montón de copias más que si no lo hago", dijo. "John Harris es un nombre familiar en los hogares [de Atari]. Entre los propietarios de ordenadores Atari, probablemente un porcentaje mayor ha oído hablar de John Harris que de la mayoría de las estrellas del rock”.

Pero ahora que la llegada de Dick Sunderland era inminente, Ken esperaba que el poder de los programadores se redujera. Ahora era un hacker convencido de que había que reprimir a los hackers. Contaba con que Dick consiguiera reducir el canon estándar de los programadores del treinta al veinte por ciento. “No creo que se necesite un genio de la programación” para crear un juego de éxito, dijo Ken. “Los días en los que se necesitaba un programador con una nota sobresaliente para escribir un juego aceptable no han terminado, pero dentro de un año habrán terminado. Los programadores no son un céntimo la docena, pero son 50.000 la docena. Mover la nave espacial [en la pantalla de vídeo] ya no es un problema. Lo que se necesita es adivinar lo que quiere el mercado, acceder a los canales de distribución, dinero, trucos, promoción de marketing”.

Sentado en su oficina ese día, hablando con su sorprendentemente sincero tono de “qué demonios”, adivinando que su compañía “tendrá 200 millones de dólares en ventas para 1985 o estará en quiebra”, añadiendo “no me preocupa mucho cuál de las dos cosas”, Ken Williams prometió retirarse a la montaña, como un peregrino de alta tecnología, y contemplar el siguiente paso para hacer realidad el milenio de la informática.

Pero para sorpresa de casi nadie, Ken Williams no cumplió su promesa de “despedirse”. Habría sido tan fuera de lugar como un hacker que abandona un programa de juego de moda antes de que se hayan escrito en él todas las características adecuadas. Ken Williams le había presentado la compañía a Dick como si su objetivo –llevar una compañía a un punto en el que fuera lo suficientemente grande como para dejarla en manos de un gerente– estuviera cumplido. Pero como un hacker, Ken Williams no veía las cosas en términos de objetivos. Todavía estaba enamorado del proceso de dirigir On-Line, y el choque de culturas entre el informalismo hacker y la rigidez burocrática sumió a la compañía en el caos.

Era casi como si se estuviera librando una lucha por el alma de la industria. Una de las primeras cosas que Dick Sunderland intentó imponer en Sierra On-Line fue una estructura corporativa rígida, una jerarquía en la que los empleados y los autores sólo podrían plantear problemas a sus superiores inmediatos. Dick pidió a las secretarias que distribuyeran copias del organigrama, con una caja en la parte superior para Ken, otra debajo para Dick y una serie de cajas debajo, todas conectadas por líneas que representaban los únicos canales de comunicación autorizados. Que este enfoque fuera antitético al hackerismo no perturbó a Dick, quien sentía que las actitudes de los hackers casi habían llevado a la empresa a la quiebra y la ruina.

Dick quería especialmente que se pusiera fin al campamento de verano. Había oído historias sobre los tumultos, las drogas, las fiestas improvisadas, las bromas durante las horas de trabajo... ¡Incluso había oído a los conserjes decir que había habido sexo real en la oficina por la noche! Ese tipo de cosas tenían que terminar. En particular, quería que Ken mantuviera una relación más ejecutiva con sus empleados y que promoviera líneas de comunicación más ordenadas y racionales. ¿Cómo se puede mantener una estructura jerárquica cuando el director ejecutivo se mete en su jacuzzi con empleadas de bajo nivel?

Para Dick, el flujo de información debe canalizarse con discreción, con una interpretación inequívoca controlada por la gente de arriba. La gente que no tiene una visión amplia de las cosas no debería molestarse por recibir información a cuentagotas. Sin embargo, con lo que Dick tuvo que lidiar en On-Line fue con un increíble remolino de rumores, alimentado por el flujo desenfrenado de información al que la empresa estaba acostumbrada. Y Ken Williams, dijo Dick, "alimenta [el

remolino de rumores] en lugar de sofocarlo. ¡No tiene sentido de la discreción!" Todo era de dominio público para Ken, desde su vida personal hasta su cuenta bancaria.

Sin embargo, Dick estaba convencido de que Ken sabía que On-Line necesitaba una gestión responsable, o moriría. Pero Ken era muy reacio a dar un paso atrás. Sunderland podría resolver la situación del personal, traer candidatos cuidadosamente considerados, mantener la nómina bajo control... y luego Ken le diría: ¡bang!, que acababa de contratar a alguien para que fuera su asistente administrativo, una vacante que no existía hasta ese mismo minuto. “¿Y a quién contrató?”, decía Dick. “Un tipo que conducía un camión de Pepsi en Los Ángeles”.

“Esto es un caso típico”, decía Dick. Recordaba haber leído sobre ello en la escuela de negocios: un empresario que se pone en marcha con una idea brillante, pero no puede manejarla cuando el negocio se vuelve grande. Todo surgía de los orígenes hacker de la empresa. Ken estaba diciendo que el tiempo de los hackers había terminado; quería limitar el poder de los programadores en la empresa. Pero no se lo estaba poniendo fácil a Dick.

Fue particularmente difícil tratar de negociar la reducción de las regalías del treinta al veinte por ciento cuando los programadores tenían la impresión de que la empresa estaba nadando en dinero. En realidad no era así, pero nadie lo creyó cuando vieron que la manteca manchando el techo. Todos sabían acerca de la casa que Ken estaba construyendo fuera de la ciudad. Tendría cuatrocientos pies de largo. Un salón de fiestas que sería el más grande de la zona. Un equipo de más de una docena estaba trabajando a tiempo completo en ella… Habían construido una oficina entera en el lugar de trabajo, con conexiones telefónicas y todo. La casa no estaba ni a medio terminar, y Ken ya había invitado a toda la compañía a que viniera al lugar los fines de semana para jugar en la cancha de racquetball incorporada. No era la mejor manera de convencer a los programadores de optar por la austeridad.

El punto de vista de Ken Williams era algo diferente. Había contratado a Dick y a menudo lo defendía, pero creía que era necesario mantener su posición. Ken se sentía responsable ante la gente que había contratado y ante la visión de la propia compañía. Conocía la industria tan bien como cualquiera; Dick era un recién llegado a la familia. Además, Ken Williams se estaba divirtiendo demasiado: irse ahora sería como alejarse de una mesa de dados cuando estás en la mejor racha de tu vida. O, más concretamente, era como decirle a un hacker que ya no podía jugar con la máquina. Esas palabras no les gustaban a los hackers. Una vez que tenías el control, el poder divino que proviene del dominio de la programación, no querías soltarlo.

Roberta Williams estaría de acuerdo. Del mismo modo que Ken trataba a On-Line como un programa informático complejo que se podía hackear, Roberta pensaba en la empresa como un proyecto creativo que debía embellecerse con cariño y estructurarse con elegancia, como un juego de aventuras. Como los autores de un juego de aventuras, ella y Ken habían disfrutado del máximo control sobre la empresa; entregarla era difícil. Comparó la situación con la contratación de una institutriz: “Uno podría pensar, ¿no sería fantástico que alguien viniera a cuidar a los niños todos los días mientras yo hago lo que quiero hacer? Puedo diseñar juegos de aventuras. Pero entonces ella empieza a decirles a los niños todo lo que pueden hacer: ‘Oh, sí, pueden comer un sándwich de mermelada’. Y puede que yo no crea en la mermelada. Puede que prefiera que coman carne. Esa persona dice: ‘La mermelada es buena, hay mucha proteína en la mermelada. Me contrataste, déjame hacer mi trabajo’. Eso es lo que nos estamos encontrando con Dick. Él dijo: “Tú me diste el

poder para hacer esto, tú querías ir y programar". Ahora decimos: "Sí, eso es lo que pensábamos que queríamos, pero resulta que no queremos ceder el control".

• • • • • • • •

Mientras que la dirección de Sierra On-Line luchaba por encontrarse a sí misma, los hackers de tercera generación estaban desanimados por los cambios en su empresa. Hablaban mientras comían cenas congeladas en Hexagon House antes de jugar a Dungeons and Dragons. O discutían el estado moral en deterioro de la empresa mientras comían pizza y Coca-Cola en Danny's, un desolado restaurante de carretera en la Ruta 41 con mesas estilo picnic cubiertas con manteles de plástico a cuadros. La mayoría de los clientes eran familias locales a las que no parecía gustarles demasiado la gente de Internet, pero era casi el único lugar de la ciudad donde se podía pedir una pizza y jugar a videojuegos, a los que los hackers jugaban compulsivamente, sin ningún sentido visible de participación o incluso interés, mientras esperaban su comida.

Estaban orgullosos de sus puestos y casi desconcertados por su buena suerte al recibir un pago por un trabajo que amaban. A principios de los años 80, hackear juegos era prácticamente la única forma de arte comercialmente viable en la que, casi sin capital, uno podía ser verdaderamente un autor: podía concebir, guionar, dirigir, ejecutar y pulir una obra sin ayuda de nadie, completando un objeto de arte que era tan bueno como el juego más vendido del mercado. Esta Tercera Generación se encontró en una posición artísticamente privilegiada. El hecho de que los editores compitieran por sus productos hacía que las cosas fueran agradables por un lado, pero confusas por el otro. No había reglas para este tipo de cosas. Era raro encontrar un hacker de veinte años con la perspicacia empresarial y la fortaleza intestinal necesarias para enfrentarse a un negociador tan enérgico como Ken Williams o tan formalmente intimidante como Dick Sunderland. Como el dinero no era el principal problema para los hackers, aceptaban casi cualquier cosa si pensaban que era justa. Los negocios no eran tan divertidos como hackear.

Aun así, en el otoño de 1982, los programadores más creativos eran los que impulsaban la industria. Brøderbund estaba en su apogeo con Choplifter, escrito por un ex hacker de inteligencia artificial de veintiocho años llamado Dan Gorlin. El juego se basaba en la crisis de los rehenes en Irán: un helicóptero cruzó las líneas enemigas e intentó rescatar a sesenta y cuatro rehenes, pequeñas figuras animadas que saludaban con la mano cuando veían el helicóptero. Era el gran juego del año y era coherente con el elegante enfoque de los Carlston en el negocio. Adoraban a sus hackers. Hablaban todo el tiempo de lo grandes artistas que eran sus "diseñadores de juegos".

Sirius había estado desarrollando sus propias superestrellas, pero Gebelli, el diseñador que había hecho casi todos sus juegos en el primer año de existencia de Sirius, no era una de ellas. Según Jerry Jewell, Gebelli pensaba que Sirius no era la mejor agencia para la exposición y venta de sus obras de arte (y esto después de recibir un cuarto de millón de dólares en su primer año, señaló Jewell con incredulidad) y, junto con un ejecutivo desertor de Sirius, comenzó su propia empresa, modestamente llamada Gebelli Software. No se unió a los puestos más altos de la industria.

Sirius sobrevivió a la pérdida importando hackers adolescentes de otras partes del país, y ellos entregaron algunos juegos de éxito llamados Beer Run, Twerps y The Earth Dies Screaming. Jerry Jewell actuó como un hermano mayor a veces ruidoso para sus jóvenes programadores. Lo que Jewell realmente ansiaba era el mercado masivo de VCS, y después de firmar un importante acuerdo para desarrollar juegos para la nueva división de videojuegos de Twentieth Century Fox, estaba entusiasmado con visiones de sus productos como nombres familiares, no solo en el mundo

de Apple o Atari, sino en todas partes. Calculó que algunos de sus programadores podrían ganar hasta un millón de dólares al año.

En On-Line, donde el VCS había sido un simple coqueteo, Ken Williams y Dick Sunderland no hablaban de un millón de dólares al año para sus programadores. Estaban tratando de reducir las regalías del treinta al veinte por ciento. Y cuando los hackers de On-Line se reunían en lugares como Danny's, comparaban notas y veían que estaban de acuerdo: el treinta por ciento era justo y el veinte por ciento no. Brøderbund y Sirius seguían ofreciendo regalías más altas. Algunos de los hackers habían sido contactados por una nueva y emocionante empresa llamada Electronic Arts. Estaba formada por ex empleados de Apple que prometieron tratar a los hackers como héroes culturales, como estrellas de rock.

Ken y Dick habían tratado de convencerlos de que el veinte por ciento era una cifra justa a la luz del drástico aumento de los costos de promoción, prueba y distribución de un juego en esta nueva etapa más profesional de la industria. On-Line estaba aumentando su publicidad, contratando más personal de soporte, reforzando su personal de promoción. Pero los programadores veían a Sunderland y su régimen como burocracia, a la que, como hackers, tenían una alergia genérica. Echaban de menos los días de campamento de verano y los apretones de manos para los contratos. A John Harris, por ejemplo, le irritaba la idea de pagar a un abogado para que le ayudara a negociar un contrato de seis cifras (“¡Cobran cien dólares sólo por leerlo!”, aullaba). Harris y los demás hackers de On-Line veían cómo se contrataba a todos esos directivos y personal de apoyo sólo para hacer lo mismo que hacía la empresa antes: lanzar los juegos que los hackers escribían. Desde su punto de vista, parecía indicar otro pecado de los hackers: la ineficiencia. Junto con un énfasis en el atractivo del marketing en lugar de la sustancia del hackeo.

Por ejemplo, On-Line gastó mucho dinero en nuevas cajas coloridas para empaquetar sus juegos, pero no consideró adecuado incluir el nombre del programador en el paquete. Ken había pensado que bastaba con dar ese crédito sólo en el manual de instrucciones que venía dentro de la caja. “Los autores deberían darse cuenta de que esto nos dará más dinero para publicidad y regalías”, dijo. Era indicativo de un nuevo “profesionalismo” en el trato con los autores.

Pero al escuchar las conversaciones en Danny’s durante el otoño de 1982, estaba claro que una atmósfera propicia para el hacking era mucho más importante para esos programadores que un manto de “profesionalismo”. Y el consenso era que casi todos los programadores estaban pensando en irse.

Incluso si Ken Williams era consciente de un posible éxodo de programadores, el problema parecía preocupar poco al fundador de la empresa. Williams estaba ocupado contratando un personal de programadores muy diferente de los posibles desertores. Impaciente con los hackers que habían llegado a él con sus habilidades en lenguaje ensamblador y hábitos de trabajo desiguales completamente formados, Ken decidió probar una fuente alternativa: utilizaría el poder mesiánico de la computadora para crear gurús de la programación donde no los había. Después de todo, los hackers ahora irritables que se quejaban de los recortes de regalías habían llegado a él con, como mucho, la experiencia de un juego o dos. Ahora sentían que les debía el mundo. ¿Por qué no encontrar gente antes de ese primer juego, gente que tuviera algunas habilidades de programación pero que aún no fueran magos del lenguaje ensamblador, y dejar que se desarrollaran bajo su supervisión? Seguramente no serían tan desagradecidos como para dejarlo por una oferta aleatoria de otra firma. Pero más importante aún, este tipo de contratación audaz estaría en consonancia con

la visión que Ken tenía para su empresa: el lugar donde el futuro de la informática llega a la gente, mejorando sus vidas.

Estableció la antigua oficina de On-Line encima de la tienda de ventas de televisores en la Ruta 41 como una oficina especialmente para programadores internos. Algunas de las personas que trabajaban allí eran programadores que cobraban regalías a quienes Ken les había ofrecido espacio para vivir gratis, como Chuck Bueche, un programador de veintiún años que conducía a las Sierras desde Texas en un viejo Jaguar XKE y que escribía bajo el seudónimo de computadora de "Chuckles". A Dick le gustaba una parte específica del primer juego de Chuckles, una persecución en un laberinto llamada Creepy Corridors: el grito penetrante y horrible que se escucha cuando el hombrecillo que estás moviendo a través del laberinto es atrapado por el monstruo que lo persigue. Considerando las capacidades de sonido relativamente mediocres del Apple, el grito fue todo un logro. Chuckles había gritado el grito más espantoso que pudo en una grabadora y había utilizado un analizador digital para imprimir cinco largas páginas de datos que, al introducirlos en la Apple, pulsaban con precisión la ubicación de la memoria para duplicar el grito. Se necesitaba casi una quinta parte de la memoria disponible de la máquina, pero para Chuck valía la pena. Los programadores más puros de On-Line estaban horrorizados por la ineficiencia.

Sin embargo, algunos de los programadores más nuevos estaban tan atrasados que Chuck, que cuestiones como esa les resultaban casi incomprensibles. Las cualificaciones de estos recién llegados iban desde títulos universitarios en informática hasta la pasión por drogarse y jugar a videojuegos. Dos eran estudiantes de ascendencia japonesa a los que Ken había contratado porque alguien le había dicho que los orientales eran trabajadores increíblemente dedicados. Algunos se sintieron atraídos por las excelentes pistas de esquí del cercano Badger Pass. Otros esperaban convertir juegos de On-Line de una máquina a otra durante el día y piratear The Great American Computer Game por la noche. En total, en el espacio de unos pocos meses Ken había contratado a casi una docena de programadores inexpertos, no hackers, por salarios de ganga, con la esperanza de que crecieran tan rápido como lo hacía la industria.

De todos los nuevos programadores de Ken, ninguno ejemplificó tanto su celo por reformar vidas mediante el poder de las computadoras como Bob y Carolyn Box. Bob Box tenía cincuenta y tantos años: habían vivido en la zona durante más de una década y trabajaban en su casa estilo rancho a ocho kilómetros de Oakhurst, en la casi indetectable aldea de Ahwahnee. Bob, que tenía cabello oscuro, ojos conmovedores y una nariz de proporciones de basset hound, medía aproximadamente un metro veinte de altura. Había sido neoyorquino, ingeniero, piloto de carreras, jockey y campeón del Libro Guinness de los récords mundiales en búsqueda de oro. Carolyn Box medía un poco más de un metro y medio, tenía el pelo largo y castaño y un atractivo que denotaba cansancio por el mundo, y era la actual campeona del Libro Guinness de los récords mundiales en la búsqueda de oro. Se habían casado veintiséis años atrás, cuando Carolyn tenía quince. Durante los últimos años, habían estado dirigiendo un negocio de suministros para la prospección de oro y buscando oro en el río Fresno, que corría en su patio trasero. La zona de Oakhurst-Coarsegold estaba en el borde sur de la veta madre de California, y el oro que los Box extraían del río (una mañana encontraron dos mil dólares en media hora) financiaba sus cursos de programación en una escuela de oficios de Fresno.

Se habían dado cuenta de que el oro de los años 80 sería el software, y su objetivo era trabajar en On-Line. Aunque Carolyn Box había estado aprensiva ante la idea de trabajar con un ordenador, comprendió al instante el concepto requerido, como si los ordenadores fueran un idioma que siempre hubiera estado hablando. Era casi sobrenatural. Ella fue la primera en la historia de la

escuela en obtener un promedio de 4.0 en sus cursos. Bob también lo hizo bien: la programación era como buscar oro, se dio cuenta: se avanzaba en pasos lógicos y se concentraba mientras lo hacía.

Pero cuando se presentaron ante Ken, él se mostró escéptico. Les dijo que los programadores normalmente alcanzaban su máximo potencial a los diecinueve años y estaban acabados a los veinte; incluso Ken, a los veintiocho, estaba prácticamente acabado. (No es que lo creyera.) Sin embargo, Ken quería darles una oportunidad a las Cajas, porque encajaban perfectamente con el sueño que tenía sobre On-Line y el gran futuro de las computadoras. Así que les dijo que pusieran algo en la pantalla usando lenguaje ensamblador, en treinta días. La escuela de las Cajas les había enseñado a programar en lenguajes de alto nivel en computadoras mainframe; no sabían nada sobre el lenguaje ensamblador de Apple. Pero trabajando día y noche, se les ocurrió un programa de 82 líneas solo cinco días después. Movía un punto por la pantalla. Ken les pidió que probaran algo diferente y, trabajando nuevamente casi todas las horas que estaban despiertos, las Cajas crearon un programa de 282 líneas con un pequeño avión que se movía por la pantalla de alta resolución. Ken las contrató y las puso a trabajar en la programación de un proyecto favorito suyo, un juego educativo.

Pronto las Cajas se pusieron a trabajar duro para conseguir que un perrito, al que llamaron Dusty como su propio perro, caminara por la pantalla. Con orgullo explicaban a los visitantes que su truco utilizaba una técnica sofisticada llamada "exclusive-or-ing", que permitía una animación sin parpadeos. Sentían que habían dado vida a Dusty Dog. "Este perro es como nuestra mascota", decía Carolyn Box. Cuando Ken vio por primera vez a Dusty Dog moverse por la pantalla, con sus patitas de basset moviéndose con una fluidez constante y sin parpadeos, casi estalló. "Son días como este los que te hacen sentir orgulloso de estar en este negocio", les dijo. Incluso estos buscadores de oro de mediana edad podían ser superestrellas del software... y Ken era el Moisés que los guió a la prometida Tierra de las Computadoras.

• • • • • • • •

Para Roberta Williams, todo representaba algo: la rehabilitación de las Cajas, los esfuerzos de Ken en pro de la comunidad, su propio ascenso al rango de diseñadora de juegos más vendida, la gran colaboración de Dark Crystal con Henson Associates, los esfuerzos artísticos de sus superestrellas del software y, sobre todo, la fantástica forma en que las computadoras habían nutrido lo que era una operación familiar en un dormitorio hasta convertirse en una empresa de 10 millones de dólares al año que pronto emplearía a más de cien personas. Ella consideraba que su historia era inspiradora. Decía mucho sobre el poder de la computadora y las vidas diferentes y mejores que las personas llevarían con la computadora. En los dos años de crecimiento de On-Line, Roberta había perdido parte de su timidez, cambiándola un poco por un orgullo feroz por sus logros. "Míranos", decía a veces en la conversación, en parte con incredulidad y en parte como una carta del triunfo para todo uso. “La gente me pregunta”, dijo en el otoño de 1982, “‘¿No te quedas sentada y dices ‘¡Guau!’? ¿No te hace sentir algo?’ La respuesta es que estamos tan constantemente asombrados que es casi un estado mental”.

Roberta quería que el mensaje de On-Line se extendiera por todo el mundo. Insistió en que On-Line contratara una empresa de relaciones públicas de Nueva York para promocionar no sólo los programas, sino también a la gente que estaba detrás de ellos. “Los programadores, los autores, van a ser los nuevos artistas del futuro”, explicó. “Puede que sea presuntuoso decir que pueden ser nuevos Robert Redfords... pero hasta cierto punto [serán] idolatrados. Los héroes del mañana”.

Dick Sunderland no compartía el entusiasmo de Roberta por la empresa de relaciones públicas de Nueva York. Había salido de una industria donde los programadores eran anónimos. Le preocupaba que los programadores de On-Line se hicieran grandes con toda esa atención. Ya es bastante duro tratar con un joven de veinte años que gana cien mil dólares al año. ¿Pueden imaginarse lo duro que será después de que aparezca en la revista People, como le sucedería a John Harris ese invierno?

La atención comenzaba a encontrar su camino hacia la misteriosa empresa de software cuyo membrete todavía llevaba la dirección de la casa de madera con estructura en forma de A de los Williams desde la que habían dirigido la empresa cuando era una operación de dos personas. Mudge Ranch Road, Coarsegold, California. El mundo quería saber: ¿qué clase de locura informática se había apoderado de ese lugar en el interior del país y qué clase de millones se estaban ganando allí, en Mudge Ranch Road? No había ningún tema más candente en los medios que los ordenadores a principios de los años 80, y con la firma de relaciones públicas de Nueva York ayudando a canalizar a los deslumbrados inquisidores, un flujo constante de llamadas telefónicas de larga distancia e incluso visitantes de larga distancia comenzaron a llegar a Oakhurst ese otoño.

En este caso, un equipo de cámaras de la revista NBC voló a Oakhurst desde la ciudad de Nueva York para documentar esta próspera empresa de la era informática para su programa de revista en vídeo. La NBC filmó las imágenes necesarias de Roberta trazando un nuevo juego de aventuras en su casa, Ken repasando sus mensajes telefónicos, Ken y Roberta visitando el sitio de construcción de su nuevo hogar. Pero el productor de la NBC estaba particularmente ansioso por hablar al corazón de la empresa: los jóvenes programadores. Jóvenes prodigio que escriben juegos y se hacen ricos. Los programadores, los internos y los que trabajan por regalías, estaban debidamente reunidos en la oficina de programación.

El productor de la NBC, con su pelo gris, bigote tupido y ojos brillantes, parecía un pregonero de feria que conoce los trucos del truculento, pero que ha mantenido la compasión. Instó a los programadores a jugar en las terminales para que su equipo pudiera filmar una toma panorámica de una próspera fábrica que medía la producción por líneas de código informático. Uno de los hackers comenzó inmediatamente a inventar un programa para crear una flor de veintiún lados en la pantalla, un programa que implicaba retener el valor de pi hasta el sexto decimal. Incluso después de que el equipo de la NBC terminara la toma inicial, el programador adolescente se sintió obligado a terminar el hack de la pantalla.

El productor estaba entrevistando a uno de los jóvenes prodigio de veintiún años de Ken.

“¿Hacia dónde va la industria?”, le preguntó solemnemente.

El joven prodigio miró fijamente al productor. “No tengo ni idea”, dijo.

Capítulo 19. Applefest

La Tercera Generación vivió con compromisos en la Ética Hacker que habrían hecho que gente como Greenblatt y Gosper se estremecieran de horror. Todo provenía del dinero. El resultado final de la programación estaba ineludiblemente ligado al resultado final en la hoja de contabilidad de un editor. La elegancia, la innovación y la pirotecnia de la codificación eran muy admiradas, pero un nuevo criterio para el estrellato hacker se había deslizado en la ecuación: cifras de ventas impresionantes. Los primeros hackers podrían haber considerado esto como una herejía: todo el software, toda la información, debería ser gratis, argumentaban, y el orgullo debería invertirse en cuántas personas usan tu programa y cuánto les impresiona. Pero los hackers de Tercera Generación nunca tuvieron el sentido de comunidad de sus predecesores, y pronto llegaron a ver las cifras de ventas saludables como esenciales para convertirse en ganadores.

Uno de los compromisos más onerosos en la Ética surgió del deseo de los editores de proteger sus cifras de ventas. Se trataba de manipular deliberadamente los programas informáticos para impedir que los usuarios pudieran copiarlos fácilmente, tal vez para distribuirlos sin pagar más al editor o al autor. Los editores de software llamaban a este proceso “protección contra copias”, pero un porcentaje sustancial de verdaderos hackers lo llamaban guerra.

Un aspecto crucial de la ética hacker era el hecho de que los ordenadores, por naturaleza, no consideran que la información sea propiedad de terceros. La arquitectura de un ordenador se beneficiaba del flujo de información más fácil y lógico posible. Alguien tenía que alterar sustancialmente un proceso informático para que los datos fueran inaccesibles para ciertos usuarios. Con un breve comando, un usuario podía duplicar un disquete “desprotegido” hasta el último byte en aproximadamente treinta segundos. Esta facilidad era espantosa para los editores de software, que se ocupaban de ello “protegiendo contra copias” los discos: alterando los programas mediante rutinas especiales que impedían que el ordenador actuara de forma natural cuando alguien intentaba copiar un disco. Un obstáculo digital que no aumentaba el valor del programa para el usuario, sino que beneficiaba al vendedor del programa.

Los editores tenían motivos legítimos para recurrir a medidas tan poco estéticas. Su sustento estaba invertido en software. No se trataba del MIT, donde el software estaba subvencionado por alguna institución. No había ARPA que pagara la cuenta. Tampoco era el Homebrew Computer Club, donde todo el mundo intentaba que le fabricaran su hardware y donde el software lo escribían aficionados y luego lo intercambiaban libremente. Se trataba de una industria, y las empresas se arruinarían si nadie compraba software. Si los hackers querían escribir juegos gratis y repartirlos a sus amigos, era asunto suyo. Pero los juegos publicados por On-Line, Brøderbund y Sirius no eran simplemente aviones de papel de la verdad lanzados al viento para difundir el evangelio de la informática. Eran productos. Y si una persona codiciaba un producto de cualquier tipo en los Estados Unidos de América, tenía que buscar en el bolsillo billetes verdes plegables o una tarjeta de crédito de plástico para poseerlo.

Esto volvía locos a los editores, pero algunas personas se negaban a reconocer este simple hecho. Encontraron formas de copiar los discos, y lo hicieron. Estas personas eran, en su mayoría, hackers.

Los usuarios también se beneficiaban de romper los discos. Algunos de ellos podían recitar una lista de justificaciones, y se oían recitarlas como una letanía en reuniones de grupos de usuarios, en tiendas de informática, incluso en la columna de cartas de Softalk. El software es demasiado caro. Sólo copiamos software que de todos modos no compraríamos. Sólo lo hacemos para probar

programas. Algunas de las justificaciones eran convincentes: si un disco estaba protegido contra copia, un propietario legítimo no podría hacer una copia de seguridad en caso de que el disco se dañara. La mayoría de los editores de software ofrecían un disco de repuesto si les enviabas un original destrozado, pero eso normalmente costaba más y, además, ¿quién quería esperar cuatro semanas por algo por lo que ya habías pagado?

Pero para los hackers, romper la protección contra copia era tan natural como respirar. Los hackers odiaban el hecho de que los discos protegidos contra copia no se pudieran alterar. Ni siquiera se podía mirar el código, admirar los trucos y aprender de ellos, modificar una subrutina que te ofendía, insertar tu propia subrutina... No se podía seguir trabajando en un programa hasta que fuera perfecto. Esto era inconcebible. Para los hackers, un programa era una entidad orgánica que tenía una vida independiente de la de su autor. Cualquiera que pudiera contribuir a la mejora de ese organismo de lenguaje de máquina debería ser bienvenido a intentarlo. Si sentías que los misiles de Threshold eran demasiado lentos, deberías ser bienvenido a examinar el código y profundizar en el sistema para mejorarlo. La protección de copia era como una figura de autoridad que te decía que no entraras en una caja fuerte que contiene cosas buenas del lenguaje de máquina... cosas que necesitas absolutamente para mejorar tus programas, tu vida y el mundo en general. La protección de copia era un matón fascista que decía: "No toques". Por principio, al menos, los discos protegidos contra copia deben estar "rotos". Del mismo modo que los hackers del MIT se sintieron obligados a comprometer la "seguridad" en la máquina CTSS, o se dedicaron a romper candados para liberar herramientas. Obviamente, derrotar al matón fascista de la protección contra copias era una vocación sagrada y sería muy divertido.

Las primeras variedades de protección contra copias implicaban rutinas de "desplazamiento de bits" que cambiaban ligeramente la forma en que la computadora leía la información de la unidad de disco. Eran bastante fáciles de vencer. Las empresas probaron esquemas más complicados, todos ellos descifrados por hackers. Un editor de software renegado comenzó a vender un programa llamado Locksmith, diseñado específicamente para permitir a los usuarios duplicar discos protegidos contra copias. ¡Ya no era necesario ser un hacker, o incluso un programador, para romper la protección contra copias! El editor de Locksmith aseguró a Apple World que su intención, por supuesto, era solo permitir a los usuarios hacer copias de seguridad de los programas que habían comprado legalmente. Insistió en que los usuarios no estaban necesariamente abusando de su programa de tal manera que los editores estuvieran perdiendo ventas. Y Buckminster Fuller anunció que se convertiría en pateador de los New York Jets.

La mayoría de los editores calculaban que habían perdido más de la mitad de su negocio a causa de los piratas de software (Ken Williams, con su característica hipérbole, calculó que por cada disco que vendía, cinco o seis eran pirateados), por lo que la protección anticopia era muy arriesgada. Curiosamente, la mayoría de las empresas contrataban como especialistas en protección anticopia al mismo tipo de joven hacker que, por lo general, pasaba horas pensando en contramedidas para romper la rutina de protección de otros. Este fue el caso de Sierra On-Line. Su encargado de la protección anticopia era Mark Duchaineau. Tenía veinte años y, durante algún tiempo, durante el gran Applefest de San Francisco de 1982, mantuvo como rehén a esta empresa que facturaba diez millones de dólares al año.

Mark Duchaineau era otro hacker de Tercera Generación que se había dejado seducir por los ordenadores. Tenía el pelo castaño que le caía magníficamente por la espalda. Sus ojos azules brillaban con una intensidad que insinuaba un fuego furioso bajo su actitud casi orientalmente

tranquila, fuego que fácilmente podía llevarlo a cometer actos inexplicables. Había fusionado su sensibilidad con la computadora en la escuela secundaria Castro Valley (California). “Tenían un teletipo”, explicaría más tarde. “Después de la escuela, me quedaba muchas horas. Me dejaban programar. Nunca fui popular, solo un solitario. [Otros] niños se interesaban por el béisbol o lo que fuera, yo estaba en la ciencia y las matemáticas. [No tenía] amigos muy cercanos; no me importaba. Era realmente interesante poder enseñarle a una máquina cómo hacer cosas. Te comunicas con la máquina... es como tratar con otra persona. Existe todo este otro universo en el que casi vives cuando estás programando. Y cuando te metes en esto siendo joven como yo, sientes una unidad con la computadora, casi como si fuera una extensión de ti mismo. Cuando imprimo comentarios en mi código, digo cosas como 'Hacemos esto, hacemos aquello...' Es como nosotros”.

Sin acceso a la computadora, dijo más tarde Mark Duchaineau, “habría habido un gran vacío... “Sería como si no tuvieras la vista ni el oído. El ordenador es como otro sentido o parte de tu ser”.

Al llegar a este descubrimiento a finales de los setenta, Mark pudo acceder a los ordenadores para su uso personal y convertirse en un hacker de la tercera generación. Mientras todavía estaba en la escuela secundaria, consiguió un trabajo en la tienda Byte Shop en Hayward. Le encantaba trabajar en la tienda de ordenadores. Hacía un poco de todo: reparaciones, ventas y programación para el dueño de la tienda, así como para los clientes que necesitaban programas personalizados. El hecho de que no ganara más de tres dólares por hora no le molestaba: trabajar con ordenadores era suficiente salario. Siguió trabajando en la tienda mientras asistía a Cal State en Hayward, donde avanzó sin esfuerzo en los cursos de matemáticas e informática. Se trasladó a Berkeley y se sorprendió por el rigor del plan de estudios de informática allí. Había desarrollado una actitud de hacker: podía trabajar intensamente durante largos períodos de tiempo en cosas que le interesaban, pero tenía poca paciencia para las cosas que no le interesaban. De hecho, le resultó prácticamente imposible retener lo que él llamaba “las pequeñas cosas que sabía que nunca necesitaría” que, por desgracia, eran esenciales para el éxito en el departamento de informática de Berkeley. Así que, como muchos hackers de tercera generación, no obtuvo el beneficio del hackeo de alto nivel que se practicaba en las universidades. Abandonó sus estudios en busca de la libertad que le proporcionarían los ordenadores personales y volvió a Byte Shop.

En la tienda se reunía un intenso círculo de piratas. Algunos de ellos incluso habían sido entrevistados en un artículo sobre piratería de software en Esquire que los hacía parecer héroes. En realidad, Mark los consideraba una especie de hackers al azar. Sin embargo, a Mark le interesaban los tipos de descubrimientos necesarios para romper la protección de copia y era bastante competente en romper discos protegidos contra copia, aunque en realidad no necesitaba los programas que contenían los discos. Como estudioso de la ética hacker, no le gustaba demasiado la idea de ser una persona que escribe esquemas de protección de copia.

Pero un día Mark estaba jugando con el sistema operativo de Apple. A menudo lo hacía, el hacker ordinario se la pasa pululando dentro del sistema. “Mi gran pasión es el descubrimiento”, explicó luego. Trabajando con computadoras, siempre podría desenterrar algo nuevo y obtener increíbles satisfacciones de sus descubriemientos. Mark estaba intentando averiguar qué activaba y desactivaba la unidad de disco en el sistema operativo y pronto supo qué la activaba, qué la hacía girar, qué hacía funcionar el cabezal y qué movía el motor. Mientras experimentaba con variaciones de las formas habituales de hacer funcionar la unidad de disco, se dio cuenta de que estaba ante un gran descubrimiento: una nueva forma de poner información en un disco. El plan de Mark implicaba organizar los datos en trayectorias en espiral en el disco para que no se pudiera acceder a

la información de forma concéntrica, como una aguja que sigue un disco, sino en varias trayectorias en espiral. Por eso Mark llamó al plan "Spiradisk". El sistema diferente frustraría los programas que rompían la protección anticopia y permitían a los piratas copiar discos. Aunque no era totalmente a prueba de piratas (nada lo es), el plan de Mark desafiaría a Locksmith y a cualquier otro plan comercial. Y llevaría muchísimo tiempo hasta para que un hacker dedicado lo descifrara.

A través de un amigo que estaba trabajando en un juego para On-Line, Mark conoció a Ken Williams. Ken sólo expresó un vago interés en el plan de Mark y durante los meses siguientes hablaron de ello por teléfono. Ken siempre parecía encontrar fallos en el sistema de Mark. Por un lado, el plan de Mark consumía demasiado espacio en un disquete. Spiradisk sólo permitía poner la mitad de la información que normalmente cabría en un disco.

Mientras arreglaba eso, Duchaineau se le ocurrió otra revelación, que le permitió no sólo almacenar la cantidad total de información en un disco, sino también acelerar el proceso por el cual el ordenador y la unidad de disco intercambiaban información. Al principio, Duchaineau dudó de que pudiera hacerse. Pero como cualquier buen hacker, lo intentó y, tras varias horas intensas de hacking, levantó la vista, estupefacto, y dijo: «Vaya, esto funciona».

Según los cálculos de Duchaineau, el proceso Spiradisk funcionaba veinte veces más rápido que el sistema operativo normal de Apple. Eso significaba que se podía cargar la información de un disco a la memoria del ordenador en una fracción de tiempo. Era revolucionario, realmente asombroso. Mark Duchaineau no entendía por qué Ken Williams se mostraba tan reacio a utilizarlo.

Ken veía cierto valor en el sistema de Duchaineau, pero no quería arriesgar toda su empresa en un plan no probado, ideado por algún joven genio cualquiera. En sus dos años como director de On-Line, Ken ya había visto muchos de ellos: verdaderos magos que eran conceptualistas brillantes, pero hackers en el peor sentido; gente que no podía terminar. ¿Qué garantía tenía de que Duchaineau pudiera —o quisiera— arreglar los errores graves que inevitablemente aparecerían en un plan tan revolucionario? Sin embargo, Duchaineau le impresionó lo suficiente como para pedirle que fuera a Oakhurst para hacer una protección anticopia más convencional. Mark, molesto por el rechazo de Ken a Spiradisk, dijo que no lo creía.

“¿Cuánto quieres que te paguen?”, le preguntó Ken.

Mark Duchaineau había estado viviendo en casa y trabajando en la tienda de computadoras por tres dólares la hora. Se arriesgó y dijo: “Diez dólares la hora”, principalmente porque, como dijo más tarde, “eso me pareció una buena cifra”.

“Bueno”, dijo Ken, “¿qué tal si te dejo vivir en una de mis casas y te doy 8,65 dólares la hora?”.

Trato hecho.

Básicamente, Ken quería un sistema de protección anticopia bastante confiable que funcionara con Form Master, una gran máquina copiadora de discos que On-Line había comprado para producir productos. ¿Podría Mark idear un programa que pudiera hacer eso? Sí. En media hora, Duchaineau ideó un plan y se puso a escribir código para las siguientes veinticuatro horas, terminando con un esquema de protección completo que, según él, “no era increíblemente confiable, no era de muy alta calidad, pero funcionaba, si [tenías] unidades de disco limpias y velocidades de disco normales”. Durante los siguientes meses, Mark lo utilizó para proteger unos veinticinco productos.

También se convirtió en el Dungeonmaster oficial de un juego de Dungeons and Dragons en funcionamiento en Hexagon House. Construida como una casa familiar suburbana tradicional, la casa estaba empezando a mostrar algo de desgaste por el abandono de una lista cambiante de hackers. Las paredes, las barandillas de madera y los gabinetes de la cocina tenían un aspecto maltratado y marcado por la guerra. Nadie se había molestado en conseguir muebles, y la habitación principal solo tenía una mesa de comedor de fórmica y sillas de cocina baratas, un juego arcade de duelo con espadas de seis pies de alto y un gran televisor en color sin soporte, conectado a un Betamax que parecía reproducir constantemente Conan el Bárbaro. En las noches de D&D, algunos de los programadores se reunían alrededor de la mesa, mientras Mark se sentaba con las piernas cruzadas sobre la alfombra sucia de pared a pared, rodeado de guías de D&D de tapa dura para dirigir partidas. Tiraba dados, prediciendo siniestramente que esta persona... o troll, según fuera el caso... tenía un cuarenta por ciento de posibilidades de ser alcanzado por un rayo lanzado por un mago llamado Zwernif. Tiraba un dado de dieciocho caras, lo miraba y miraba hacia arriba con una expresión de desconcierto esos ojos intensos y decía, ansioso por la siguiente crisis: "Todavía estás vivo". Luego hojeaba el libro en busca de otra confrontación de vida o muerte para los jugadores. Dirigir una partida de D&D era un gran ejercicio de control, al igual que las computadoras.

Mark siguió presionando a favor de Spiradisk. Su entusiasmo por implementar el plan difícil de descifrar no se debía a un deseo de frustrar a los posibles piratas; Duchaineau consideraba que era un sacrificio llevar a cabo su plan maestro más altruista. Esperaba que Spiradisk generara suficientes regalías para que él pudiera comenzar su propia compañía, una que no se guiaría por los estándares improductivos del comercialismo, sino por los objetivos progresistas de la investigación y el desarrollo. La compañía de Duchaineau sería un paraíso para los hackers, con programadores que tendrían todas las herramientas imaginables a su disposición para crear un software asombroso. Si un programador sentía que la compañía necesitaba un equipo, digamos un osciloscopio supercalibrado, no tendría que obtener permiso de canales de administración desconectados... . él y sus compañeros hackers tendrían una gran influencia en el proceso. Inicialmente, la compañía de Mark escribiría software de última generación; el propio Mark soñaba con escribir la versión definitiva para computadora de Dungeons and Dragons.

Pero el software era sólo el comienzo. Una vez que los ingresos pudieran sostenerlo, la compañía de Mark entraría en el hardware. El objetivo final sería crear un ordenador lo suficientemente bueno como para manejar un juego de arcade tan bien como los juegos de monedas más sofisticados. Tendría un sintetizador de música incorporado mejor que los modelos actuales más avanzados; tendría potencia más que suficiente para ejecutar el "entorno" de software soñado de Mark llamado SORDMASTER (Screen Oriented Data Manipulation System), que sería como tomar el mejor programa que se ejecuta hoy y extender su valor a la décima potencia... un ordenador, en palabras de Mark, que "haría todo lo que quisieras".

Finalmente, Ken Williams aceptó permitir que el Dungeonmaster protegiera contra copias los programas de On-Line con Spiradisk. Mark recibía cuarenta dólares por hora por configurar las cosas, cinco mil dólares al mes por mantener el sistema y un uno por ciento de regalías sobre todos los discos que utilizaban su sistema. Mark también lo arregló para que lo primero que un usuario viera cuando iniciara un Spiradisk fuera el nombre de la "empresa" de Mark, Bit Works.

Como Ken sospechaba, había problemas con el plan. A menudo había que reiniciar los discos una o dos veces antes de que el programa se cargara correctamente. Williams empezó a desencantarse con Duchaineau. En opinión de Ken, Mark era una de esos brillantes pero desenfocados hackers prima

donnas. Ken creía que Mark era capaz de dar un golpe que podría resultar crítico para toda la industria: crear un formato de disco que admitiera a Apple, Atari e IBM en el mismo disco, en lugar del sistema actual, que requería un disco separado para ejecutarse en cada máquina. "Mark sabe cómo hacerlo", se quejaba Ken. "Podría hacerlo en seis semanas. No quiere hacer el esfuerzo. Es trabajo. Se sentó, trabajó durante una semana y perdió el interés en el proyecto. Puede hacerlo, pero no le entusiasma. No es divertido". Según Ken Williams, "Tendrías que ser suicida para dejar que tu empresa dependa de un tipo como Duchaineau". Cuando le señalaron a Ken que su empresa dependía de ese hacker de tercera generación, Ken Williams admitió que ese era el caso.

Esto se puso claramente de manifiesto en el Applefest anual en San Francisco. Uno de los puntos culminantes de ese gran evento de fin de semana, un bazar en el que todas las empresas que vendían productos para Apple exhibirían y venderían sus productos, iba a ser la presentación de una secuela largamente esperada y profusamente aumentada de uno de los juegos de Apple más queridos de todos los tiempos, Ultima. En un tremendo golpe, On-Line Systems había conseguido el juego y su voluble autor, que escribió bajo el seudónimo de Lord British.

El Ultima original era un juego de rol de fantasía en el que el jugador creaba un personaje, le asignaba ciertos "puntos de atributo" en áreas de resistencia, sabiduría, inteligencia, destreza y fuerza y, viajando por un planeta misterioso, buscaba mazmorras y torres, iba a aldeas en busca de suministros y chismes útiles, y luchaba contra elfos, guerreros y magos. Aunque el juego estaba escrito en BASIC y se ejecutaba bastante lento, era una proeza magistral de imaginación y fue un éxito de ventas de Apple. Pero cuando Lord British preparó su secuela, dejó en claro que deseaba dejar a su editor actual, quien, según dijo, no le estaba pagando todas sus regalías.

Recibió un aluvión de ofertas de empresas de software. Aunque solo tenía veinte años en ese momento, Lord British no era ajeno a las situaciones de presión: su verdadero nombre era Richard Garriott y era hijo del astronauta de Skylab Owen K. Garriott. Había conocido y disfrutado del reflejo de la fama de su padre, especialmente cuando su Skylab 2 estaba en el aire y la familia parecía ser el foco de atención del mundo. Richard había crecido en la zona de la bahía de Nassau, con una gran cantidad de ingeniería, en Houston y se había involucrado en computadoras en la escuela secundaria, donde convenció a sus profesores para que le permitieran tomar clases particulares de programación. Su plan de estudios era escribir juegos.

En muchos aspectos, era un chico americano bien adaptado. Por otro lado, se quedaba despierto toda la noche con el ordenador Apple en su dormitorio. "Cuando salía el sol me daba cuenta de lo tarde que era y me desmayaba en el acto", explicó más tarde. Hacía tiempo que le interesaban los juegos de rol de fantasía y le fascinaba especialmente la cultura medieval, y pertenecía a un club llamado la Sociedad de Anacronismos Creativos. Cuando era estudiante de primer año en la Universidad de Texas, se unió al equipo de esgrima, pero en realidad le gustaba mucho más la lucha con espadas, balanceándose libremente, subiéndose a una mesa, al estilo de Errol Flynn. Quería fusionar sus dos intereses e intentó crear un juego de ordenador que lo hiciera. Después de escribir durante meses, completó su vigésimo octavo juego y lo llamó Alkabeth, y se quedó atónito cuando un editor que vio por casualidad una de las copias que Richard envió a sus amigos de forma gratuita se ofreció a publicarlo y enviarle dinero. ¿Por qué no? Pidió el seudónimo de Lord British porque una vez unos chicos de un campamento de informática se burlaron de él diciendo que parecía que venía de Inglaterra (no era así).

Alkabeth ganó suficiente dinero para varios estudios universitarios. Su siguiente juego, Ultima, era más ambicioso y con sus regalías de seis cifras compró un coche, abrió grandes cuentas Keogh y IRA e invirtió en un restaurante de Houston. Ahora estaba considerando bienes raíces.

Garriott vio su continuación como algo especial. Había aprendido lenguaje de máquina especialmente para escribirlo y estaba mareado con el nuevo poder que le daba: sentía que le permitía ver la memoria, el microprocesador, los circuitos de vídeo... entendías lo que hacía cada bit y adónde iban las líneas de datos. Y la velocidad que te daba era increíble. Sólo con este poder pudo hacer realidad Ultima 2, porque en Ultima 2, Richard Garriott estaba escribiendo una verdadera epopeya, una que permitía al jugador hacer más de lo que cualquier otro jugador de un juego de ordenador había hecho antes. Insistió en que algunas de estas habilidades se enumeraran en la caja en la que se vendía el programa:

- Capturar barcos
- Secuestrar aviones
- Viajar por todo el sistema solar
- Enfrentarse a transeúntes inocentes
- Ser perseguido por agentes de la KGB
- Ser abordado en callejones oscuros
- Luchar contra piratas en alta mar
- Ser seducido en un bar
- Cenar en tu restaurante favorito
- Conocer a personas prominentes dentro de la industria informática
- Lanzar hechizos mágicos a criaturas malvadas
- Visitar el castillo de Lord British
- Explorar mazmorras oscuras y mortales
- Robar a mercaderes
- Matar criaturas viciosas
- Atravesar fuerzas impenetrables
- Desarrollar la magia más poderosa conocida por el hombre

Garriott había encarnado la metáfora de la computadora (crear y poblar un universo privado) en un juego que permitía al jugador vivir en el mundo de la imaginación de Lord British. Al mover el personaje que habías creado, designando rasgos de personalidad, conseguías poderes, herramientas, medios de transporte, armas... y entre los orcos asesinos y los magos malvados, también podías encontrarte con personajes basados en personas reales, muchos de ellos amigos de Richard Garriott, personajes que, en consonancia con sus personalidades reales, te darían información críptica que te ayudaría a resolver el enigma.

Richard Garriott podía haber mostrado ambición y complejidad joyceanas, pero admitía que carecía de dotes literarias: «No sé deletrear, no tengo técnicas gramaticales y he leído menos de veinticinco

libros en mi vida». Esto le avergonzó al principio, pero ahora se decía a sí mismo que el ordenador era una forma artística viable. Y al vender Ultima 2 a un nuevo editor, su principal preocupación, además de una tasa de regalías no negociable del treinta por ciento, era que el paquete y el marketing fueran artísticamente coherentes con el virtuoso programa informático que contenía. Para ello se necesitaba una caja grande ilustrada profesionalmente, un mapa del universo en tela con líneas que marcaban distorsiones temporales, tarjetas de cartón especiales que contenían las docenas de comandos disponibles para los jugadores y un manual elaborado y de gran tamaño en el que cada una de las dieciséis páginas parecía un documento de piel de oveja descolorida.

Ninguna de estas exigencias disuadió a los editores de software de intentar contratar a este hacker tan rentable. Ken Williams lo persiguió sin descanso, oliendo a best seller. Después de llevar al joven autor en avión a Oakhurst, aceptó todas las exigencias de Lord British, incluso el treinta por ciento de regalías. Ken Williams quería que firmara en ese momento y, como dijo Garriott más tarde, "se enojó mucho por el hecho de que no iba a firmar nada [ese día]". Pero después de regresar a Texas, Garriott firmó. "No pude ver ninguna razón para no hacerlo".

Ahora, después de meses de retraso, algunos debido a un período de depuración inesperadamente largo (nunca ha habido un período de depuración inesperadamente corto en la historia de las computadoras), algunos debido al hecho de que los mapas de tela fueron encargados a una firma de Irán, que de repente se cerró al comercio estadounidense tras la crisis de los rehenes, el programa estaba completo.

Garriott tenía el juego en la mano en Applefest; adornado con cadenas de oro y una túnica de gamuza y cuero, el alto texano de cabello castaño y rasgos angulosos atrajo a multitudes al stand de On-Line mientras desvelaba su obra maestra. La gente no podía creer su buena suerte mientras se reunía alrededor de Garriott, de veintiún años, que demostraba con naturalidad cómo podrían encontrar una ocasión en Ultima 2 para viajar a Plutón. ¡Éste es el tipo que escribió Ultima! Los pedidos pendientes del programa de 59,95 dólares se contaban por decenas de miles; Richard Garriott esperaba que el primer cheque de regalías por Ultima 2 fuera mayor que la suma de los cheques que había cobrado anteriormente por la autoría del juego. Habría sido un joven muy feliz, de no ser por este problema que impedía que Ultima 2 se lanzara ese mismo fin de semana. El problema era Mark Duchaineau. No había protegido el programa contra copia y no estaba claro que lo hiciera.

El Dungeonmaster le había insistido a Dick Sunderland que su sistema Spiradisk funcionaría perfectamente en Ultima 2, acelerando el tiempo de carga y ralentizando sustancialmente la red pirata que esperaba ansiosamente el desafío de descifrarlo. Desestimó los problemas anteriores de On-Line con Spiradisk como insignificantes. Insinuó que podría haber algunos problemas al proteger el programa contra copia sin Spiradisk. Dick sospechaba que los argumentos de Mark estaban motivados por su afán por promocionar Spiradisk y cobrar las regalías, que valdrían más de diez mil dólares en un éxito de ventas como Ultima 2.

Richard Garriott, su amigo y compañero programador Chuck Bueche y el gerente de productos de On-Line concluyeron conjuntamente que Spiradisk sería demasiado arriesgado, Dick Sunderland llamó a Duchaineau para decirle que protegiera el programa contra copia a la antigua usanza. Pero Mark seguía siendo evasivo.

Dick estaba furioso. Esta extraña criatura, este megalómano Dungeonmaster de veintiún años, que vivía en una de las casas de Ken, aprovechándose de la reputación de On-Line para promocionar su

sistema... ahora tenía el descaro de insinuarle a Dick que el programa más lucrativo de la temporada no se lanzaría, ¡porque quería protegerlo contra copias a su manera! Por aterradora que sonara su amenaza, Mark, como la única persona que protegía las copias, tenía el poder para respaldarla: llevaría semanas traer un reemplazo. Lo que era aún más aterrador era que Mark Duchaineau, si así lo decidía, ¡podría retener sus servicios para toda la línea de productos de On-Line! La compañía no podría lanzar ningún producto sin él.

Sunderland estaba desconcertado. Ken no había llegado al Festival. Todavía estaba de regreso de Chicago, donde había asistido a la convención de fabricantes de pinball y videojuegos con monedas. Dick ni siquiera tenía los medios técnicos para juzgar la validez de las afirmaciones de Duchaineau. Así que reclutó a uno de los jóvenes programadores de On-Line, Chuck Bueche, para que fuera a la larga fila de teléfonos públicos que había a la entrada del Applefest y llamara a Duchaineau (sin dejar entrever que era a instancias de Dick, por supuesto) para entender los tecnicismos involucrados. Tampoco estaría mal que el programador suavizara la línea dura del Dungeonmaster.

De hecho, aunque Bueche era un agente doble incómodo, la llamada pareció romper el atolladero. Tal vez lo que hizo que Duchaineau cediera fue que la llamada le recordó que estaba ralentizando un proceso que eventualmente permitiría a los usuarios beneficiarse del triunfo de un compañero programador: Mark Duchaineau estaba en el incómodo papel de un hacker que intentaba impedir que se publicara el programa valioso de otro hacker. En cualquier caso, aceptó proteger el producto contra copias, aunque cuando Ken Williams se enteró del incidente, su respeto por el hacker Mark Duchaineau se hundió aún más. Más tarde juró que expulsaría a Duchaineau de Oakhurst con alquitrán y plumas, tan pronto como On-Line pudiera averiguar cómo reemplazarlo.

• • • • • • • •

Durante dos años, el espectáculo Applefest había sido la principal reunión de las empresas del mundo Apple como On-Line, Sirius, Brøderbund y docenas de proveedores de software, placas complementarias y periféricos que funcionaban con Apple. Era un momento para celebrar la máquina que había dado a la Hermandad su sustento e inspiración, y las empresas estaban más que felices de entretener a los miles de propietarios de Apple que se sumergían ansiosamente en un mar de juegos de arcade, buffers de impresora, unidades de disco, guías de programación, joysticks, tarjetas RAM, monitores RGB, simulaciones de guerra y estuches rígidos para transportar computadoras. Era el momento de renovar los lazos dentro de la Hermandad, de buscar nuevos programadores, de redactar los pedidos, de dejar que la gente viera quién eras y cómo dirigías tu propio espectáculo.

Pero el Applefest de San Francisco de 1982 resultaría ser el último de los Applefest importantes. Por un lado, On-Line y sus competidores estaban lanzando programas para varias máquinas; Apple ya no era dominante. Además, las empresas empezaban a ver los espectáculos abiertos a los usuarios como una pérdida de tiempo, energía y dinero, recursos que podrían gastarse en lo que se estaba volviendo esencial: las grandes Ferias de Electrónica de Consumo de Las Vegas y Chicago, que se estaban convirtiendo en las más importantes, el gran evento exclusivo para profesionales, el de la electrónica de consumo, en el que el héroe no era el hacker, sino el hombre que registraba las ventas.

Aun así, el espectáculo estaba repleto, otro indicio de la explosión económica que había llegado a los ordenadores. En medio del estruendo de pies arrastrados, voces y ruidos de juegos electrónicos, lo que surgió fue una melodía de prosperidad sin precedentes. Casi en todas partes había millonarios

atendiendo stands, millonarios que hace sólo dos años estaban sumidos en actividades oscuras y no rentables. Luego estaban las empresas emergentes, con stands más pequeños o sin stands, soñadores atraídos por el emocionante y afrodisíaco aroma que emanaba del mundo Apple y el mundo relacionado de los ordenadores domésticos.

Ese olor a éxito estaba volviendo loca a la gente.

La gente intercambiaba historias increíbles, y hasta la saga de alta tecnología más sorprendente de Horatio Alger era superada sin esfuerzo por un ejemplo aún más sorprendente del boom. Era una fiebre del oro, pero también era cierto que la inversión mínima para los buscadores serios era una suma más formidable que cuando Ken Williams empezó. El capital de riesgo era una necesidad, obtenido de los hombres con trajes de raya diplomática que cenaban en los mediocres restaurantes franceses del Valle, pronunciaban koans de En busca de la excelencia en los seminarios del sector ("Marketing, Marketing, Marketing") y se referían solemnemente a sí mismos como "tomadores de riesgos". Eran personas intolerables, oportunistas del sueño hacker, pero si lograbas que te guiñaran el ojo, las recompensas podían ser infinitas. Nadie lo sabía mejor que la gente del Applefest que estaba trabajando para crear una empresa llamada Electronic Arts. Su idea era pasar por alto lo que consideraban las prácticas ya anticuadas de las empresas de la Hermandad y establecer una firma que fuera incluso más nueva que la Nueva Era. Una empresa que llevara el software a otro ámbito por completo.

Electronic Arts había definido su misión en un pequeño folleto dirigido a los "artistas del software" que estaban tratando de atraer para que se alejaran de sus editores actuales. Este prospecto sonaba como algo escrito por un redactor de anuncios que había fusionado con éxito las sensibilidades de los trajes de tres piezas y la droga hawaiana de primera calidad. Estaba repleto de párrafos de una sola frase que contenían palabras como "emoción", "visión" y "no tradicional". Su verdadero brillo residía en el enfoque de su atractivo: estaba dirigido directamente al corazón hacker de sus lectores. Electronic Arts sabía que no debía avivar el factor avaricia prometiéndoles a los hackers suficientes regalías para comprar Trans-Ams color cereza y citas caribeñas con apasionados fanáticos del software. En cambio, se confesaba: "Creemos que es más probable que los autores innovadores provengan de personas que son independientes y no trabajan en una 'fábrica' de software o en la 'burocracia'". Prometía desarrollar herramientas y utilidades fantásticas y poderosas que estarían disponibles para los autores de EA. Se comprometía a mantener el tipo de valores personales que los hackers aprecian más que el dinero. El resultado sería una "gran empresa de software", lo que implicaba que, en lo que se refiere a programadores creativos, honestos, con visión de futuro y con valores hacker, no existía en la actualidad una empresa de ese tipo.

Electronic Arts fue la idea original de Trip Hawkins, que había dejado su trabajo como director de marketing de Apple para el proyecto USA para dedicarse a ello. Fundó la empresa en una habitación adicional en la oficina de una empresa de capital de riesgo. Hawkins reunió a un equipo de Apple, Atari, Xerox PARC y VisiCorp y, en un golpe de efecto que seguro encantaría a cualquier hacker, consiguió que Steve Wozniak aceptara formar parte de la junta directiva.

Electronic Arts no tenía un stand en el Applefest, pero su presencia se hizo sentir. Organizó una gran fiesta la noche de la inauguración y su gente trabajó en el recinto ferial como políticos. Una de ellas, una ex ejecutiva de Apple llamada Pat Mariott, una mujer alta, delgada y rubia con enormes gafas redondas y un bronceado intenso, explicaba con entusiasmo la empresa a un periodista. Trip fundó Electronic Arts, dijo, porque vio lo rápido que empezaba a funcionar el negocio y "no quería

perder la oportunidad". Pat lo acompañó porque lo vio como una oportunidad para divertirse y, no por casualidad, ganar dinero.

"Quiero hacerme rica, por cierto", dijo, explicando cómo, en Silicon Valley, la riqueza era omnipresente. Adondequiera que miraras, veías sus artefactos: BMW, opciones sobre acciones y, aunque no lo mencionó, cocaína en cantidades ingentes. Tampoco se trataba de una riqueza común y corriente de cientos de miles de dólares al año; se trataba del modo Creso, donde la aritmética de punto flotante apenas era suficiente para contar los millones. Cuando veías a tus amigos entrar en el negocio, pensabas: ¿Por qué yo no? Así que, cuando se abría una ventana hacia la riqueza, saltabas por ella de forma natural. Nunca ha habido una ventana tan atractiva como la de la industria del software. Pat Mariott lo resumió en un susurro, citando al periodista gonzo Hunter S. Thompson: "Cuando las cosas se ponen raras, los raros se vuelven profesionales".

Pat Mariott esperaba entrar en acción en modo Creso sin comprometer sus valores personales de los años sesenta. Por ejemplo, nunca trabajaría para una empresa despiadada. Pat había sido programadora, había experimentado la cultura hacker en Berkeley y el entorno profesional de la malvada IBM. "Berkeley era verdad y belleza. IBM era poder y dinero. Yo quería ambas cosas", dijo. Electronic Arts parecía el camino. Los productos y la filosofía de la empresa serían verdad y belleza, y los fundadores de la empresa serían todos poderosos y ricos. Y los programadores, que serían tratados con el respeto que merecían como artistas de la era informática, serían elevados a la categoría de estrellas de rock o de cine.

Este mensaje logró abrirse paso en el Applefest, lo suficiente como para que grupos de programadores comenzaran a reunirse fuera del Convention Hall para tomar los autobuses que supuestamente los llevarían al Stanford Court Hotel, donde Electronic Arts estaba celebrando su gran fiesta. Un extraño grupo incluía, entre otros, a varios programadores de On-Line y a John "Captain Crunch" Draper.

John Draper, cuyo pelo oscuro y fibroso volaba en todas direcciones, había logrado buenos resultados por sí solo. Durante su estancia en prisión, después de que lo pillaran usando la interfaz del teléfono de Apple como una caja azul, había escrito un programa de procesamiento de textos llamado "Easy Writer", que le permitió ganar una suma considerable. Sorprendentemente, cuando IBM buscó un programa para publicar como su procesador de textos oficial, eligió Easy Writer; la empresa que publicó el programa de Draper tuvo el buen sentido de actuar como intermediaria con IBM, sin dejar entrever que el autor del programa era el famoso Capitán Crunch. Se decía que Draper había ganado un millón de dólares con la transacción. Nadie lo habría adivinado por sus vaqueros descoloridos, su viejo polo y su aparente necesidad de hacerse un tratamiento dental. Mark Duchaineau lo miró con una mezcla de asombro y repulsión mientras el ex phreaker lo sermoneaba sobre algún aspecto técnico del IBM PC.

Pronto, dejaron de lado el autobús y llamaron a un taxi. El taxista cometió el error de fumar. John Draper casi le arrancó el cigarrillo de la boca al conductor, exigiendo a todo pulmón que se abrieran todas las ventanillas del taxi en la fría y húmeda noche de noviembre de San Francisco.

El hotel era bastante elegante, y los hackers, con vaqueros y zapatillas de deporte, parecían intimidados. Sin embargo, Electronic Arts se había preparado para ellos: junto con una banda de rock que tocaba música de baile, la empresa había alquilado más de una docena de videojuegos de pago por monedas, adaptados para ofrecer juegos gratuitos ilimitados. Allí fue donde se dirigieron inmediatamente los hackers. A medida que la fiesta se calentaba, era evidente que habían aparecido

muchos de los autores más importantes de la industria, algunos para comprobar las cosas, otros genuinamente interesados en esta aventura más nueva que la nueva era.

Sin embargo, el centro de atención era el miembro de la junta directiva de EA, Steve Wozniak, citado en una serie de discursos como "el hombre que lo empezó todo". Era un epíteto que habría perseguido a algún joven genio ansioso por sacudirse el pasado y pasar a cosas más nuevas, pero Wozniak parecía deleitarse con él; Hacía más de un año que viajaba por todo el país a reuniones de la industria, recibiendo los mismos elogios. Había gastado una parte considerable de su dinero en Croesus Mode en presentar festivales de rock masivos. Todavía creía fervientemente en la ética hacker, y dondequiera que iba no sólo predicaba ese evangelio, sino que se ponía a sí mismo como ejemplo de ello. Esta noche, por ejemplo, predicó a un pequeño grupo sobre los males del secretismo, utilizando la política actual de Apple como principal ejemplo. El secretismo y la burocracia asfixiante allí eran tales que no estaba seguro de si alguna vez volvería a la empresa construida sobre su creación, el Apple II.

En conjunto, la fiesta fue un éxito, crepitante con la dulce sensación de que todos estaban montados en la cresta de una ola gigantesca. ¿Ocurrieron cosas así en los primeros días de Hollywood? ¿En la industria discográfica en los años sesenta? El futuro estaba a sus pies, una mezcla de hackerismo y riqueza incalculable, y la impresión general era que allí se estaba haciendo historia.

Los hackers de On-Line se marcharon impresionados. Algunos de ellos firmarían con Electronic Arts en los meses siguientes. Y uno de ellos se marchó con una sonrisa particularmente satisfecha: había conseguido las puntuaciones más altas en Pac-Man, Robotron y Donkey Kong. Para un autor de éxito, una noche para recordar.

• • • • • • • •

Ken Williams llegó al Applefest de mal humor. La convención de fabricantes de pinball en Chicago había sido frustrante; las empresas gigantes, en particular Atari, habían tirado montones de dinero a los fabricantes de máquinas tragamonedas para conseguir los derechos de primera opción para la versión para ordenador doméstico de cualquier juego que fuera mínimamente jugable. Una repetición de Frogger, que Ken había adquirido por una mera comisión de regalías del diez por ciento, estaba fuera de cuestión.

Ken, que viajaba con Roberta, fue directo al expositor de su empresa en el Applefest. On-Line había ocupado un enorme stand, situado justo en la entrada, junto a las escaleras mecánicas que llevarían a las masas hasta el complejo subterráneo Brooks Hall. El stand exhibía un mural fotográfico gigante de una cascada de la Sierra, enfatizando el cambio de nombre de On-Line Systems. El stand también tenía muchas combinaciones de computadora, joystick y monitor incrustadas dentro de paneles para que las hordas de jóvenes fanáticos de las computadoras pudieran jugar los últimos juegos de Sierra On-Line. Los monitores estaban colocados en los paneles muy por encima del nivel de los ojos, para que los espectadores pudieran apreciar fácilmente cuán hábilmente estaban elaborados los juegos. Y para atraer clientes al stand, una enorme pantalla de proyección de televisión en color estaba conectada a una computadora que reproducía continuamente el juego más vendido de On-Line, Frogger. Como la versión de Apple no tenía la música continua y los gráficos de nivel arcade de la versión Atari de John Harris, los empleados de On-Line escondieron discretamente una computadora Atari 800 debajo de una cortina, y estaban ejecutando esa versión en el Applefest: el equivalente a exhibir un automóvil japonés en una exhibición de General Motors. Con todas esas multitudes, todo ese alboroto, ¿quién se daría cuenta?

Dos personas que se dieron cuenta fueron Al y Margot Tommervik, editores de Softalk: se dieron cuenta de inmediato porque Frogger no era simplemente otro programa de On-Line para ellos. Representaba un giro deprimente de los acontecimientos. Como todos los que habían visto la brillante conversión de John Harris, se habían sentido sobrecogidos y encantados cuando lo vieron a principios de ese año. Pero cuando vieron la contraparte de Apple poco después, se quedaron impactados. Era horrible. Para Al y Margot, los miserables gráficos de esta versión de Frogger representaban en el mejor de los casos un error, y en el peor, una traición absoluta al mercado de Apple, que había nutrido a On-Line en primer lugar.

El mundo de Apple era un coto espiritual para los Tommerviks, y parecía que al hacer un Frogger de Apple inferior, On-Line había escupido despectivamente en el suelo de este coto exaltado. Obviamente, Al y Margot le debían al resto del mundo de Apple hacer algo que rara vez hacían en su revista: dar a un juego una crítica negativa. El crítico que asignaron estuvo de acuerdo con los Tommerviks y escribió una descripción mordaz del juego: “Tiene casi tanta alma como una lechuga de un mes en el Sahara”, escribió. “Tu rana se parece a un peón de ajedrez con alas vestigiales... los troncos en el río parecen recién salidos de una fábrica de Oscar Meyer...”

El crítico no se detuvo allí. Preguntó qué había sido de la compañía que una vez se mantuvo como un “bastión de calidad en un mar de mediocridad”. Mientras que les daba a los propietarios de Atari un gran programa, On-Line les estaba dando a los propietarios de Apple “una bofetada en la cara”. Algo serio, que llega al corazón de la ética hacker, que te instruye a seguir trabajando hasta que tu hack supere los esfuerzos anteriores. “¿Nos han abandonado?”, preguntaba la crítica a On-Line.

Como Margot y Al habían sido tan cercanos a Ken y Roberta, intentaron explicar la crítica a sus amigos antes de que apareciera en el Softalk de diciembre. Pero tuvieron dificultades para llegar a Ken o Roberta. Las filas burocráticas se estaban endureciendo en On-Line; ya no había ninguno de los Williams que cogiera el teléfono. Te tocaba una recepcionista que te conectaba con una secretaria que anotaba tu nombre y el número de tu empresa y te decía que alguien te devolvería la llamada. Si tenías suerte. Finalmente, Al se puso en contacto con el hermano de Ken, John, quien le dijo que había razones por las que el juego tenía ese aspecto... pero esas razones nunca se las explicaron a los Tommervik. La gente de On-Line estaba demasiado enfrascada en las batallas de gestión como para dar explicaciones.

Al y Margot habían llevado consigo ejemplares tempranos de la revista a Applefest. Ver el truco tortuoso de On-Line con el Atari Frogger no hizo más que confirmar su creencia de que la reseña era justa. Supusieron que después de hablar con Ken y Roberta, las cosas se resolverían amistosamente. ¿No estaban todos en esto por lo mismo? ¿Para mantener el fantástico impulso humanista de Apple World? No se podía permitir que un desacuerdo como el de Frogger afectara a una misión tan importante como esa.

Cuando alguien en el stand de On-Line le dio a Ken una copia del nuevo Softalk, él se dirigió inmediatamente a la reseña de Frogger. Roberta leyó por encima de su hombro. Sabían que la reseña sería negativa y más o menos esperaban alguna crítica a los gráficos del juego, aunque no en un lenguaje tan mordaz. Pero no tenían idea de que la reseña continuaría cuestionando si la compañía, al lanzar una versión tan buena de Frogger para Atari y una versión tan lamentable para Apple, había vendido el Apple World. "O Frogger es un error o una traición", concluía la reseña. "Tendrás que tomar tu propia decisión".

"Esto va mucho más allá de lo justo", dijo Ken. Por un lado, dijo, Softalk no se dio cuenta de lo difícil que era hacer el juego en Apple en comparación con Atari. Los Tommerviks aparentemente habían decidido atacar a la compañía, todo después de que los Williams habían ayudado a sacar adelante la revista cuando la Hermandad estaba recién formándose. Roberta pensó que esta confrontación se venía gestando desde hacía tiempo: por alguna razón, Softalk siempre parecía tratar con indiferencia a On-Line. Pero cada vez que Roberta preguntaba a los Tommerviks si algo andaba mal, ellos decían que todo estaba bien.

“No nos quieren en esa revista”, le dijo Roberta a Ken. “Deberíamos retirar nuestros anuncios”.

Era otra señal de que la Hermandad no era inviolable. Las cosas eran más importantes ahora que las amistades personales. Ahora que las empresas de la Hermandad eran más como negocios del mundo real, competían entre sí. Los Williams rara vez hablaban con gente de Brøderbund o Sirius y nunca más intercambiaban secretos. Jerry Jewell lo resumió más tarde: “Solíamos socializar mucho con Brøderbund y On-Line... ahora la actitud es que si invitas a los competidores a fiestas, todo lo que van a hacer es desenterrar toda la suciedad que puedan sobre ti e intentar contratar a tus programadores. [Socializar] se vuelve cada vez menos posible a medida que el negocio se vuelve más y más despiadado. Quieres que tus competidores sepan cada vez menos lo que estás haciendo”. Era algo que tenías que aceptar.

Ken tocó esto brevemente cuando se encontró con Doug Carlston en el piso de exhibición. Doug parecía ser el que menos había cambiado: era tan sincero y abierto como siempre, el más cuerdo de la Hermandad. Ambos estuvieron de acuerdo en que debían reunirse más, como lo habían hecho en los viejos tiempos, hace un año. Hablaron de la nueva competencia, incluida una nueva empresa que estaba entrando al mercado con un capital de 8 millones de dólares. “Eso nos hace parecer juguetes”, dijo Carlston. “Nosotros obtuvimos un millón [en capital de riesgo]. Ustedes obtuvieron... ”

“Un millón dos”, dijo Ken.

“Ustedes renunciaron a más. Nosotros renunciamos al veinticinco por ciento”.

“No, renunciamos al veinticuatro por ciento”.

Hablaron sobre la ausencia de Sirius Software en el Applefest, otra indicación de que la acción estaba cambiando a ferias comerciales exclusivas. Ken pensó que el impulso de Jerry Jewell para comercializar cartuchos en masa era bueno: “Será más rico que todos nosotros”, predijo.

Doug sonrió. “No me importa si todos los demás se enriquecen... siempre y cuando yo lo haga”.

“No me importa si alguien se hace rico”, dijo Ken, “siempre que yo me haga más rico”.

Ken intentó sumergirse en el espíritu del evento y llevó a Roberta, que lucía elegante con jeans de diseñador, botas altas y boina negra, a un recorrido rápido por los stands. Ken era un conversador natural y en casi todos los stands lo reconocieron y lo saludaron calurosamente. Pidió a una media docena de jóvenes programadores que se acercaran a Oakhurst y se hicieran ricos hackeando para On-Line.

Aunque se esforzaron por evitar el stand de Softalk, los Williams se encontraron con Margot Tommervik. Después de un saludo incómodo, ella le preguntó a Ken si había visto la portada de Dark Crystal.

“Todo lo que vi fue la crítica de Frogger”, dijo. Pausa, “pensé que era un poco desagradable”.

Margot lo abrazó para no mostrar resentimientos. “Oh, Ken, el juego fue pésimo”, dijo. “Lo hicimos porque te amamos. Porque tu material es mucho mejor que eso. Esperamos más de ti”.

—Bueno —dijo Ken, sonriendo entre dientes—, ¿no te pareció que iba más allá del juego? Decía todo tipo de cosas sobre nuestra empresa.

Margot no quiso ni oír hablar de ello. Los Williams, sin embargo, no consideraban que el asunto estuviera cerrado. Para ellos, era un ejemplo más de cómo la gente cambia cuando las cosas se ponen serias.

Esa noche, On-Line organizó una cena en un restaurante italiano de North Beach. Ken llevaba semanas hablando de la posibilidad de una buena y antigua noche de alboroto en On-Line, pero aunque todo el mundo estaba de humor para celebrar, el evento nunca despegó. Tal vez porque sólo se invitó a dos programadores (Richard Garriott y Chuck Bueche) y el resto de la gente era mayor, muchos de ellos sumidos en la mentalidad de ventas, contabilidad y marketing. Hubo los brindis habituales y repetidos, y por supuesto, hubo Steel (aguardiente de menta), grandes tragos de una botella con un vertedor de bebidas de metal. Muchos de los brindis estaban dirigidos al invitado de honor, Steve Wozniak. Ken se lo había encontrado esa tarde y, para su deleite, el legendario hacker había aceptado la invitación tardía a cenar. Ken Williams se aseguró de contarle a Woz sobre su posesión más preciada, el vínculo más preciado que tenía con el espíritu de la era liberadora de la computadora doméstica: una placa madre original de Apple I. A Ken le encantaba ese trozo de epoxi y silicio; significaba algo para él que el propio Woz lo hubiera cableado a mano en un garaje, allá por la era del Pleistoceno de 1976. Woz nunca se cansaba de oír hablar de la época del Homebrew y agradecía el cumplido de Ken. Wozniak sonrió ampliamente cuando lo saludaron, esta vez Dick Sunderland. El Steel se repitió una vez más.

Para Woz, sin embargo, el momento culminante de la noche fue conocer a Lord British. Meses después, todavía hablaba de lo emocionado que había estado de hablar con semejante genio.

A la cena le siguió un agitado viaje a una discoteca en el edificio Transamerica. Después de todo ese jolgorio, Ken y Roberta estaban exhaustos cuando regresaron a su hotel. Una llamada de emergencia los esperaba. Se había producido un incendio en la casa de madera con estructura en forma de A en Mudge Ranch Road. Solo el heroísmo de una niñera había salvado a sus dos hijos. Sin embargo, la casa estaba gravemente dañada. Ken y Roberta exigieron hablar con sus hijos para asegurarse de que estaban a salvo y se dirigieron a casa de inmediato.

Ya era de día cuando llegaron al lugar donde antes estaba su casa. Los niños estaban a salvo, todo parecía cubierto por un seguro y los Williams habían planeado mudarse de todos modos al año siguiente, a la mansión palaciega que estaba en construcción. El incendio no fue la catástrofe que podría haber sido. Ken Williams tenía solo una pena persistente: la pérdida de cierto objeto material irremplazable en la casa, un artefacto que significaba algo para él mucho más allá de su mera utilidad. El fuego había consumido la placa madre Apple I de Ken, su vínculo con el comienzo idealista de la era humanista de las computadoras. Estaba en algún lugar entre los escombros, dañada sin posibilidad de reparación, para nunca ser encontrada.

Capítulo 20. Mago contra mago

En diciembre de 1982, Tom Tatum, flacucho, moreno, con bigote y tan tranquilo como su perezoso acento sureño implicaba, estaba de pie en el podio del salón de baile del Las Vegas Sands. Detrás de él, sentados incómodamente en una fila de sillas, había diez hackers. Tom Tatum, ex abogado, lobista y ayudante de campaña de Carter, ahora un proveedor líder de programación de video "docusports", pensó que había agarrado por casualidad un premio mayor más grande que el de cualquier máquina tragamonedas en el casino a solo unos metros de donde estaba parado.

"Este es el evento donde Hollywood se encuentra con la Era de la Computadora", dijo Tom Tatum a la multitud de periodistas y comerciantes de computadoras en la ciudad para el show de Comdex. "El ultra-concurso de los ochenta".

La creación de Tom Tatum se llamó Mago contra Magos. Iba a ser un concurso televisivo donde los diseñadores de juegos juegan los juegos de los demás para un conjunto de premios. Tatum había reunido a programadores de empresas como On-Line y Sirius porque percibía la llegada de un nuevo tipo de héroe, uno que luchaba con cerebro en lugar de músculos, uno que representaba la audaz voluntad de Estados Unidos de mantenerse por delante del resto del mundo en la batalla tecnológica por la supremacía: el hacker.

A diferencia de las producciones deportivas anteriores de Tom Tatum, que incluyeron el Maui Windsurfing Grand Prix de 1981 y el Telluride Aerobatics Invitational, este Wizard vs. Wizards tenía el potencial de atraer a una nueva audiencia al género de los docusports. “Solo un pequeño porcentaje de la población poseerá una bicicleta Super Cross”, explicó más tarde. “Pero cuando ves a la gente usando la computadora en casa, es increíble”.

Obviamente, las competencias que le importaban a la gente ahora se estaban llevando a cabo en las salas de juegos y frente a las computadoras Apple. Imaginen cuántos sintonizarían para ver competir a los profesionales. Es más, como dijo Tatum, “el atractivo de este programa es el doble golpe” de los propios autores –esos extraños informáticos de ciencia ficción– compitiendo entre sí.

“¡Éstas son las nuevas estrellas!”, dijo Tom Tatum en Las Vegas, pero las nuevas estrellas parecían incómodas siendo exhibidas en un escenario de Las Vegas como tantas otras deformes candidatas a Miss Universo. La belleza del hackerismo era taoísta e interna, deslumbrantemente impresionante cuando uno podía percibir la atrevida mezcla de idealismo y cerebración, pero menos que convincente cuando se presentaba como un coro en un salón de baile de Las Vegas. Las sonrisas de los hackers eran rígidas, sus trajes no les quedaban bien (aunque algunos llevaban trajes deportivos especialmente hechos –aunque todavía no les quedaban bien–). Incluso el observador más obtuso podía adivinar que la mayoría de ellos preferirían estar hackeando en casa. Pero con motivaciones mixtas de curiosidad, presión de sus editores, el deseo de pasar unos días en Las Vegas y, sí, vanidad, habían venido al Sands para competir en lo más caliente que Tom Tatum había hecho nunca, con la posible excepción, según reconoció más tarde, de las finales del Super Cross de Miller High Life.

El concurso incluiría a hackers de siete empresas. Jerry Jewell estaba en el lugar con los dos mejores jugadores de Sirius. On-Line llegaría mañana. Después de la presentación, Jewell se jactó ante uno de los competidores de que uno de sus hombres bien podría ser el mejor jugador de videojuegos del mundo. "Lo he visto jugar a Robotron durante cuatro horas", dijo.

El hacker no se dejó intimidar. "¿Ves esto?", respondió con voz chillona, extendiendo la mano. "Esta es mi ampolla de Robotron. Normalmente paro después de una hora porque mis manos son muy sensibles".

Más tarde, en su habitación de hotel, Jewell observó cómo sus hackers practicaban los juegos programados para la competición. Jewell estaba exultante por el acuerdo de su compañía con Twentieth-Century Fox Games. Los cartuchos VCS que sus programadores ahora diseñaban fueron ampliamente distribuidos y comercializados intensamente por Fox; su compañía fue la primera de la Hermandad en tener sus juegos anunciados en televisión y distribuidos en puntos de venta masivos. “Una cosa es ver tu producto Apple en la pared de una tienda de computadoras”, decía Jerry Jewell, “pero cuando ves un estante de tus cosas en K-Mart, sabes que has llegado”.

Ken Williams llegó a Las Vegas a tiempo para una reunión previa al concurso que Tatum celebró para los doce concursantes y sus patrocinadores. Después de haberse recuperado rápidamente del incendio, Williams estaba listo para ser el único competidor en el concurso que en realidad era un editor. Él y los demás sacaron sillas en un semicírculo para escuchar a Tatum describir las reglas.

“Este es un nuevo tipo de concurso”, se dirigió Tom Tatum al grupo. “No sucedería si no fuera por la televisión. Está creado para la televisión. Las reglas han sido desarrolladas para la televisión”. Tatum explicó que en este nuevo tipo de concurso había dos conjuntos de valores en conflicto: el Valor Uno era la necesidad de una competencia honesta y justa, y el Valor Dos era la necesidad de hacer todo lo posible para que las cosas se vieran bien en la televisión. Tatum dijo que ambos valores eran importantes, pero que siempre que los dos valores entraban en conflicto, él elegía el Valor Dos.

Luego Tatum describió la imagen con la que comenzaría el programa: una toma de la franja de neón de Las Vegas de noche con un mago, símbolo del hacker, asomándose. Sobre él, rayos de luz brotaban de las puntas de sus dedos. Un icono omnipotente de la Nueva Era. Esta imagen pareció impresionar a los informáticos, como también lo hizo la imagen que dibujó Tom Tatum de los beneficios de competir en un evento televisivo. Podría elevarlos, dijo Tatum, al estatus de nombres conocidos. “Una vez que este programa se lance y otros programas comiencen a suceder, comenzarán a suceder cosas”, dijo Tatum. “Puedes obtener ingresos de otras fuentes, como productos publicitarios”.

La mañana del programa de televisión, antes de que se encendieran las cámaras, el exiguo público en el Sands Ballroom pudo presenciar algo que diez o veinte años antes se habría considerado más allá de la imaginación de Heinlein, Bradbury o incluso del visionario residente del MIT, Ed Fredkin. Especialistas en maquillaje aplicaban con indiferencia maquillaje de panqueque en las caras de jóvenes programadores informáticos inquietos. La era del hacker de los medios había comenzado.

Tom Tatum había contratado a una actriz de telenovelas, peinada para matar y armada con una sonrisa de esmalte de dientes, para presentar el programa. Tuvo problemas con su primera frase sobre cómo ésta era la primera vez en la historia intergaláctica que los magos de la informática y los genios tecnológicos del mundo se reunían para competir; hicieron falta quince iteraciones antes de una toma. Sólo entonces empezó la competición, y sólo entonces quedó tristemente claro lo aburrido que era ver a un grupo de hackers sentados en largas mesas, con joysticks entre las piernas, cada uno con un pie calzado con zapatillas bajo la silla y el otro extendido bajo la mesa, la mandíbula ligeramente floja y los ojos clavados en la pantalla.

A diferencia de otras formas más atractivas de competición de vídeo, los programadores no se mostraban muy expresivos cuando tenían que limpiar una pantalla de alienígenas o ser aniquilados por un rayo púlsar vengador. Los espectadores perspicaces tenían que estar muy atentos a las muecas o a la frustración entrecerrada para saber cuándo un movimiento en falso acababa en una explosión de vídeo. Cuando los jugadores se enfrentaban a la despreciada señal de GAME OVER antes de que se alcanzara el límite de tiempo de cinco minutos, levantaban tristemente la mano para que uno de los jueces tomara nota de la puntuación. Una agonía de derrota mediocre.

Tatum pensó que esta deficiencia videogénica se remediaría con cortes rápidos, tomas de las pantallas de computadora y entrevistas concisas con los gladiadores de silicio. Las entrevistas generalmente eran como la que la estrella de telenovelas realizó con Dan Thompson, de diecinueve años, de Sirius, quien rápidamente se estableció como favorito.

> ESTRELLA DE TELENOVELAS: ¿Cómo se siente tener una ventaja tan dominante de cara a las semifinales?
>
> THOMPSON (se encoge de hombros): Genial, supongo.

¡Corten! ¿Podemos hacer esto otra vez? La segunda vez, Dan no se encogió de hombros. ¿Una vez más, por favor? A esta altura, la lógica digital y la técnica de resolución de problemas de Dan Thompson se habían aplicado al rompecabezas. Tan pronto como la pregunta salió de la boca del astro de la telenovela, se inclinó hacia el micrófono, con los ojos fijos en la cámara.

“Bueno, se siente maravilloso. Sólo espero poder continuar con esto…”. Había sintetizado las superficialidades del lenguaje de los deportistas.

Thompson, beneficiario de horas de joysticks en un teatro Time de Chuck E. Cheese Pizza en Sacramento, ganó el concurso. Ken Williams había actuado admirablemente, considerando que apenas tuvo la oportunidad de mirar algunos de los juegos antes de jugarlos; el hecho de que quedara en sexto lugar en la general era testimonio de su capacidad para llegar instantáneamente al corazón de un juego de computadora, y el hecho de que a los veintiocho años todavía le quedaban algunos reflejos.

En la suite de Tatum esa noche, el empresario de videos estaba fuera de sí. “Creo que hemos visto el evento televisivo más revolucionario en años”, dijo. Predijo que estos hackers capturarían la imaginación de Estados Unidos: atletas que no reciben una paliza física, pero emanan una intensidad fascinante. Levantó su copa de licor por el futuro del hacker como el nuevo héroe americano.

• • • • • • •

Un programador de On-Line que había dado señales de convertirse en un héroe de los medios era Bob Davis, el ex alcohólico a quien Ken Williams había elevado a la categoría de autor de juegos y considerado su mejor amigo. Williams había coescrito con Davis el juego de aventuras Ulises y el vellocino de oro, y las líneas finales de la reseña de Softalk de Margot Tommervik se leen como una justificación triunfal de la decisión de Ken Williams de asociarse con la computadora para cambiar el mundo:

> On-Line Systems tiene dos nuevos ganadores en “Ulises”: la aventura, que es la mejor de On-Line desde “El mago y la princesa”; y Bob Davis, un nuevo autor del que esperamos ver muchas aventuras nuevas.

El paquete que Sierra On-Line envió para atraer a posibles autores incluía una carta abierta de Bob Davis, quien contaba su experiencia de haber sido “picado por el virus informático”, haber visto su juego pasar por un proceso de producción sin complicaciones y haber recibido regalías “más que abundantes y siempre a tiempo”. Davis concluyó escribiendo: “Así que ahora solo paso mi tiempo esquiando en las pistas del lago Tahoe, mirando mi grabadora de video, conduciendo mi nuevo auto y viviendo bastante cómodamente en mi nueva casa de tres habitaciones. Te recomiendo encarecidamente que hagas lo mismo”.

Sin embargo, poco después de que Ken regresara de Las Vegas, Bob Davis no pudo ser localizado en las pistas de esquí, detrás del volante de su automóvil o en su nuevo hogar. Recibía visitas sólo en la cárcel del condado de Fresno. Davis vestía un mono rojo desgastado de prisión y una mirada angustiada. Tenía el pelo largo y rojo brillante, una barba roja descuidada y líneas de preocupación en su rostro que lo hacían parecer mayor de sus veintiocho años. Como el vidrio entre el prisionero y el visitante era grueso, sus conversaciones se llevaban a cabo a través de receptores telefónicos a ambos lados del vidrio.

Bob Davis no había recibido muchas visitas en sus pocas semanas en la cárcel. Había estado tratando de conseguir que Ken Williams lo rescatara, hasta ahora sin éxito. Había pasado de alcohólico a superestrella del software a convicto drogadicto, todo en meses. Había pensado que la computadora lo liberaría. Pero la computadora no había sido suficiente.

Para un desertor de la escuela secundaria convertido en borracho a quien le gustaban en secreto los acertijos de lógica, la programación había sido una revelación. Davis descubrió que podía sumergirse tanto en el mundo de las drogas que ya no necesitaba beber. Su fortuna en la empresa aumentó mientras dirigía el proyecto Time Zone, coescribió su juego de aventuras y comenzó a aprender lenguaje ensamblador para la desconcertante máquina VCS. Pero, tan de repente como su vida había cambiado para mejor, comenzó a desmoronarse.

“Tengo un poco de dificultad para manejar el éxito”, dijo. La sensación embriagadora que le producía ser un autor de software de gran éxito le hizo pensar que podía manejar el tipo de drogas que antes le habían hecho la vida miserable.

Había drogas en On-Line, pero Bob Davis no podía permitirse el lujo de consumirlas con la moderación que otros lo hacían. Afectaba a su trabajo. Tratar de aprender el código VCS era bastante difícil. Pero el rápido éxito de Davis con Ulysses, escrito en el relativamente sencillo lenguaje de desarrollo de aventuras de Ken Williams, lo había preparado para la gratificación instantánea de la programación, y se frustró. “Traté de inventar excusas”, dijo Davis más tarde. “[Dije] que On-Line se estaba volviendo demasiado corporativo para mí”. Lo dejó, pensando que escribiría juegos por su cuenta y viviría de las regalías.

Había estado trabajando en un juego de VCS, pero a pesar de las horas que pasó intentando conseguir algo de movimiento en la pantalla, no pudo. Aunque Ken Williams se dio cuenta de que Bob era el tipo de persona que conseguía sus avances solo cuando alguien lo guiaba —“Si hay alguien ahí, estará ahí [trabajando] hasta las 4 de la mañana”, comentó Ken una vez— Ken no podía tomarse el tiempo para ayudar a su amigo. Davis intentaba ponerse en contacto con Ken y decirle lo infeliz que estaba, pero Ken a menudo estaba fuera de la ciudad. Bob tomaba más cocaína, inyectándosela directamente en las venas. En desacuerdo con su esposa, se iba de casa cuando lo drogaban, añorando todo el tiempo estar en casa, de vuelta a la nueva vida centrada en la computadora que había comenzado: el tipo de vida de superestrella del software de la que había

hablado en ese testimonio en primera persona que On-Line todavía incluía en el paquete enviado a los posibles autores.

Bob Davis volvía a casa tarde por la noche, encontraba que su mujer se había ido y empezaba a llamar a todos los que conocía en On-Line, a las casas de todos los programadores, a los lugares donde sabía que ella no podía estar, con la esperanza de que alguien supiera dónde estaba. Incluso los desconocidos que contestaban al teléfono oían su voz lastimera, desgarrada por el pánico. "¿Has visto a mi mujer?" No, Bob. "¿Sabes dónde puede estar?" No la he visto, Bob. "Es muy tarde y no está en casa, y estoy muy preocupado". Estoy seguro de que volverá a casa. "Espero que esté bien", decía Bob, conteniendo los sollozos. "Nadie me dirá dónde está".

Todo el mundo se sentía fatal por Bob Davis. Fue una de las primeras cosas que le hicieron saber a Dick Sunderland que On-Line no era una empresa más en un sector más: la misma noche en que contrataron a Dick, Davis estaba en una de sus giras por Oakhurst. Allí estaba ese fantasma, esa plaga inquietante del sueño informático, esa oportunidad de oro perdida. Como una conciencia implacable, Bob Davis acosaba a sus antiguos amigos con llamadas, a menudo pidiendo dinero. El programador y testigo de Jehová Warren Schwader, a quien Bob le gustaba a pesar de sus frecuentes palabrotas y su hábito de fumar, una vez se ofreció a pagar su hipoteca directamente, y Bob, que quería dinero en efectivo, colgó el teléfono de golpe... pero más tarde convenció a Warren para que le prestara mil dólares.

Como todo el mundo, Schwader quería creer que Bob Davis podía volver a la computadora y programar para salir de su torbellino de drogas. Al final todos se rindieron. Personas como el programador Jeff Stephenson, que intentó inscribir a Bob en un programa de AA, se disgustaron cuando Bob comenzó a pasar cheques sin fondos. "Mi hábito pasó de trescientos a novecientos dólares al día", explicó Davis más tarde. "Acabé echando a mi esposa de casa. Intenté dejar el hábito dos veces", pero no pude. Le pidió a Dick Sunderland anticipos de sus regalías, y cuando Dick se negó, se ofreció a vender sus futuras regalías "por una miseria", dijo Sunderland más tarde. Pero pronto las regalías de Davis fueron directamente al banco para pagar deudas pasadas. Estaba vendiendo los muebles para conseguir más dinero para las drogas, vendió su ordenador Apple, el instrumento de magia que le había convertido en alguien.

Fue un alivio para la gente de On-Line cuando Bob Davis acabó en la cárcel. Lo arrestaron en un motel. La gente supuso que el cargo era pasar cheques sin fondos, pero el propio Davis dijo que era por cocaína y que se había declarado culpable. Quería entrar en un programa de rehabilitación de drogas para empezar de nuevo. Había estado tratando de enviar un mensaje a Ken, pero Ken pensó que Bob Davis estaba mejor en la cárcel, donde podría librarse del hábito.

El autor del duodécimo juego de ordenador más vendido del país, según la Hot List de Softsel, habló por el teléfono de la prisión y explicó cómo la había fastidiado, cómo había visto la luz deslumbrante que emitía el ordenador, se había deleitado con ella, pero no podía estar a la altura. Estaba a mitad de frase cuando el teléfono se quedó sin habla y los visitantes de la cárcel de Fresno tuvieron que volver a entrar en la noche. El visitante podía entender sus palabras mientras las gritaba en el cristal antes de que lo sacaran: "Que Ken me llame".

• • • • • • • • •

La difícil situación de Bob Davis ejemplificaba el caos que reinaba en Sierra On-Line ese invierno. A primera vista, parecía una empresa que se acercaba a la respetabilidad (los conglomerados seguían haciendo ofertas de compra, la más reciente por 12,5 millones de dólares más un contrato

de 200.000 dólares al año para Williams). Pero debajo de la apariencia de una empresa en crecimiento y próspera había dudas persistentes. Esto se acentuó con un anuncio en diciembre de 1982 de que las cifras de ventas de videojuegos de Atari habían caído en picado. La gente de On-Line y otras empresas de juegos de ordenador se negaron a ver esto como una indicación de que el campo era una moda pasajera.

La desorganización no había hecho más que aumentar con el nuevo y difícil manejo de Sierra On-Line. Por ejemplo, un juego que Dick había considerado atractivo, un juego de varios niveles con un escenario de minería, llevaba semanas languideciendo en el departamento de adquisiciones. El programador llamó para cerrar un trato y, cuando Dick consiguió seguir su rastro a través de la empresa, el estudiante universitario que había programado el juego había renunciado a On-Line y había vendido el programa a Brøderbund. Bajo el nombre de Lode Runner, el juego se convirtió en un éxito de ventas, nombrado «Juego del año 1983» por muchos críticos. La historia era un paralelo inquietante a lo que había ocurrido cuando Ken Williams había intentado vender Mystery House a Apple menos de tres años antes: la joven empresa informática, demasiado confusa en la gestión para actuar con las respuestas rápidas como el rayo quela industria informática exigía, no se decidió a expresar interés hasta que fue demasiado tarde. ¿Sierra On-Line, todavía una empresa incipiente, era ya un dinosaurio?

El conflicto por el control entre Ken Williams y Dick Sunderland había empeorado. Los nuevos empleados, orientados a las ventas, apoyaban a Dick; sin embargo, a la mayoría de los primeros empleados y a los programadores no les gustaba el presidente y sus técnicas secretas de gestión. Los sentimientos hacia Ken eran contradictorios. Hablaba del espíritu de On-Line, pero también del "crecimiento" de la empresa, como si el software informático fuera algo que requiriera una empresa dirigida de manera tradicional, repleta de planes de negocios y una burocracia rígida. Si esto era cierto, ¿qué decía esto sobre el sueño hacker de confiar en la computadora como modelo de comportamiento que mejoraría y enriquecería nuestras vidas? Era una crisis moral que perseguía a todos los pioneros de la industria que habían comenzado sus negocios pensando que la tecnología mágica que tenían para ofrecer haría que sus negocios fueran especiales. El marketing de masas se cernía ante ellos como un anillo tolkienesco omnipotente: ¿podrían agarrar el anillo y no ser corrompidos? ¿Podrían preservar el idealismo que existía en su misión? ¿Podría el espíritu del hackerismo sobrevivir al éxito de la industria del software?

Ken se preocupaba por esto: "Cuando trabajaba para Dick, me quejaba de trabajar de ocho a cinco [y no en el modo hacker despreocupado]. Ahora quiero un equipo de programación que trabaje de ocho a cinco. Es como pasar de ser un hippie a ser un capitalista o algo así. Creo que hay muchos programadores [aquí] que se sienten traicionados. Como John Harris. Cuando él llegó, era una jornada de puertas abiertas, mi puerta estaba abierta a cualquier hora. Él podía venir, podíamos hablar de técnicas de programación. Yo lo llevaba a lugares. Nunca hicimos negocios con un contrato. No lo necesitábamos. Si no confiábamos el uno en el otro, no deberíamos hacer negocios juntos. [Ahora] eso ha cambiado. Ya no sé cuáles son mis objetivos. No estoy seguro de cuál es la forma de dirigir la empresa. De alguna manera, al contratar a Dick, me eché atrás. Es la incertidumbre lo que me molesta, no sé si tengo razón o no".

Seguían ocurriendo eventos inexplicables. Como el incidente en la oficina de programación. Un joven que trabajaba horas extra dibujando imágenes en la computadora para el juego de aventuras Dark Crystal, que había sido empleado de On-Line casi desde el principio, dejó su tableta gráfica un día y comenzó a gritar, a golpear las paredes, a arrancar carteles y a agitar un cuchillo largo hacia la

joven aterrorizada que había estado dibujando imágenes a su lado. Luego agarró un perro de peluche y lo apuñaló furiosamente a un perro de peluche, lo hizo trizas y el relleno voló por la pequeña oficina. Los programadores de la habitación contigua tuvieron que detenerlo y el joven esperó en silencio hasta que se lo llevaron con calma. La explicación: simplemente perdió el control, eso fue todo.

El hacker Jeff Stephenson, que trabaja en el proyecto secreto de IBM (también retrasado), expresó la frustración general: “No sé para quién se está dirigiendo la empresa, pero no son los autores, que me parecen el pan de cada día de la empresa. La actitud es: ‘Entonces tú eres John Harris, ¿quién te necesita?’ Sí. Él había logrado hacer mucho dinero en esta empresa. Pero parecen pensar que mientras puedas conseguir un empaque elegante y etiquetas bonitas, se va a vender”.

De hecho, John Harris había notado esta tendencia. El diseñador de juegos charlatán que había escrito dos de los programas más populares en la historia de la microcomputadora se debatía entre la lealtad y el disgusto por la forma en que se ignoraba la ética hacker. A Harris no le había gustado el hecho de que los nombres de los autores no estuvieran en las nuevas cajas, y ciertamente no le había gustado cuando, después de mencionarle esto a Dick, Dick le respondió: "Espera, antes de que hagamos nada, ¿cuándo estará listo tu próximo juego para nosotros?". Todo un cambio con respecto a los días del campamento de verano. Harris creía que los tiempos en que todos dejaban de trabajar y hacían bromas, como ir a Hexagon House y poner todo en la casa, incluso los muebles, patas arriba, eran los mejores tiempos para On-Line; todos trabajaban mejor y más duro para una empresa en la que era divertido trabajar.

John Harris también estaba molesto por lo que él consideraba el retroceso de la empresa con respecto a los altos estándares artísticos. John se tomaba como una ofensa personal si la empresa lanzaba un juego que él sentía que era perjudicial para el cerebro de alguna manera. Estaba absolutamente horrorizado por las versiones de Atari y Apple de Jawbreaker 2. El hecho de que los juegos fueran secuelas oficiales de su diseño de juego original era irritante, pero a John no le habría importado si hubiera sentido que los juegos estaban ejecutados de manera soberbia. Pero no era así: las caras sonrientes eran demasiado grandes y los extremos de los conductos por los que se movían las caras de un lado a otro estaban cerrados. A John le molestaba la caída de calidad. De hecho, sentía que los juegos más nuevos de On-Line en general no eran muy buenos.

Quizás lo peor de On-Line en lo que a John se refería era el hecho de que Ken Williams y su compañía nunca se habían rendido lo suficiente ante lo que, en la mente de John Harris, era la innegable grandeza del Atari 800. Tenía una identificación salvaje con esa máquina. John llegó a la triste conclusión de que en On-Line, Atari siempre ocuparía un segundo lugar después del Apple. Incluso después de la debacle de Frogger, cuando la versión Atari de John era de última generación y la versión de Apple era relativamente un desastre, Ken no parecía tomarse en serio el Atari. Esto deprimió tanto a John Harris que decidió que tendría que dejar On-Line por una compañía que compartiera sus puntos de vista sobre el Atari.

No fue fácil. On-Line había sido bueno con John Harris. Ahora tenía una casa, respeto, periodistas de la revista People que venían a entrevistarlo, una camioneta con tracción en las cuatro ruedas, un televisor de proyección, una cuenta bancaria abultada y, después de todos esos problemas desde Fresno hasta el Club Med, John Harris ahora tenía una novia.

En una convención de ciencia ficción, se encontró con una chica que había conocido casualmente en San Diego. Ella había cambiado desde entonces: "Estaba muy bien", recordaría John más tarde.

"Perdió peso y se operó la nariz". Ahora era actriz y bailarina de danza del vientre en Los Ángeles. Incluso le habían pedido que bailara, explicó John, en el lugar de danza del vientre más prestigioso de Hollywood. "En San Diego, siempre parecía estar con otra persona; esta vez no. Me prestaba más atención a mí que a cualquier otra persona. Pasamos diecinueve de las siguientes veinticuatro horas juntos". La veía a menudo después de la convención de ciencia ficción; ella se quedaba en su casa durante semanas y él iba a Los Ángeles a verla. Empezaron a hablar de matrimonio. Era una felicidad que John Harris nunca había conocido.

Sabía que su mentor Ken Williams había sido fundamental para provocar el cambio en su vida. Parecería lógico, entonces, que John Harris, albergando estas profundas dudas sobre la empresa con la que estaba tan estrechamente identificado, hubiera llevado sus objeciones directamente a Ken Williams. Pero John Harris no se atrevía a hablar con Ken sobre lo cerca que estaba de dejar On-Line. Ya no confiaba en Ken. Cuando John intentaba explicar por qué se sentía engañado por On-Line, Ken hablaba de todo el dinero que estaba ganando. En un momento dado, Ken le dijo a un periodista de People que John ganaba 300.000 dólares al año, y cuando Harris había tratado de corregir esa cifra, Ken lo había avergonzado al darle a John su cheque de regalías más reciente. El cheque de cuatro meses (Harris cobraba mensualmente, pero a veces no llegaba a recogerlo durante un tiempo) era de 160.000 dólares. Pero ese no era el punto; Ken nunca habló del dinero que On-Line estaba ganando con el trabajo de John Harris. Sin embargo, en lugar de decírselo, John simplemente aceptaba lo que Ken proponía. No sabía si era timidez o inseguridad, o qué.

Así que no habló con Ken Williams. Visitó a su nueva novia y trabajó en un nuevo ensamblador para el Atari y visitó el salón de juegos local (logrando una puntuación alta en la máquina Stargate) y pensó en ideas para su próximo juego. Y habló con la gente de Synapse Software, una empresa que se tomó en serio el Atari 800.

De hecho, Synapse era casi exclusivamente una empresa de software para Atari Home Computer, aunque estaba planeando hacer conversiones a otros sistemas. Los juegos que produjo Synapse estaban llenos de acción, explosiones, disparos y gráficos brillantemente concebidos. John Harris los consideró increíbles. Cuando fue a visitarlos en Berkeley, le impresionó que los programadores estuvieran bien atendidos, que intercambiaran utilidades y se comunicaran mediante un tablón de anuncios informático administrado por la empresa. Cuando John Harris se enteró por un programador de Synapse de que parte de una rutina de sonido de un juego de Synapse había sido literalmente extraída del código objeto de una copia del disco Frogger robado a John en la Software Expo (ese robo que había hundido a John en su profunda y dolorosa depresión), estaba menos enojado por la violación que encantado de que un hacker de Synapse hubiera revisado su código y encontrado algo que valiera la pena apropiarse. Synapse le prometió a John que obtendría todo el soporte técnico que necesitaba; podría unirse a su comunidad de programadores. Y le ofrecieron un derecho de autor del veinticinco por ciento. En resumen, Synapse le ofreció todo a un hacker de Atari que On-Line no le había ofrecido.

John aceptó hacer su próximo proyecto para Synapse. La superestrella del software de On-Line se había ido.

John estaba sentado en su casa preguntándose cómo decirle a Ken Williams cuando sonó el teléfono. "Tierra", respondió John, como de costumbre. Era Ken. John estaba nervioso. —Ahora estoy programando para Synapse —soltó, en un tono que Ken tomó como insoportablemente arrogante. Ken le preguntó por qué, y John le dijo que porque estaban ofreciendo el veinticinco por

ciento de regalías en lugar del veinte por ciento de Ken. —Eso fue un poco estúpido —dijo Ken. Pero John tenía muchas cosas que decir. A toda prisa, empezó a decirle finalmente a Ken todas las cosas sobre On-Line que antes le había intimidado demasiado. Incluso más cosas de las que había pensado anteriormente: John se estremecería más tarde al recordarlo: decirle al presidente de la empresa que tanto había hecho por él que los productos de la empresa eran basura.

John Harris, con todos sus programas perdidos, códigos fuente extravagantes, retrasos perfeccionistas y chovinismo Atari 800, había sido el alma hacker de Sierra On-Line. Había sido a la vez la pesadilla de la existencia de Ken Williams y el símbolo de sus logros. Su cercanía con Ken había sido representativa de la nueva benevolencia que compañías como On-Line sustituirían a los abismos habituales entre jefe y trabajador. Ahora John Harris se había ido, tras haber lanzado una jeremiada sobre la forma en que On-Line había abandonado su misión original. Lo que dejó atrás fue Frogger, el programa más vendido desde hacía semanas en la Hot List de Softsel.

• • • • • • • •

Lejos de estar afectado por la pérdida de John Harris, Ken parecía eufórico después de la pérdida. Era como si no hubiera estado alardeando varios meses atrás de que el nombre de John Harris en un programa de Atari vendería juegos. Ken estaba seguro de que la era del autor independiente que pirateaba juegos había terminado: "Creo que tengo una visión de los autores que es diferente de la visión que los autores tienen de los autores, y rezo por tener razón. “Es decir, los [hackers] con los que estoy tratando ahora están en el lugar correcto en el momento correcto. John Harris estaba allí. Es un programador mediocre que no es creativo en absoluto y que estaba programando Atari en el momento correcto”.

En lugar de un hacker que pierde el tiempo tratando de hacer que un producto sea perfecto, Ken prefería programas menos pulidos que se entregaran a tiempo, para poder empezar a crear una campaña publicitaria en torno a ellos. No como Frogger, que se retrasó porque un día John Harris decidió que simplemente no quería trabajar. “No se puede dirigir un negocio con gente que se deprime cuando les roban sus cosas. Se necesitan personas que cumplan cuando dicen que lo harán, al precio que dicen que lo harán y que sean capaces de resolver sus problemas por sí solas. John Harris quiere que vayas a beber con él, que hables por teléfono, que vayas al Club Med, que le hagas sexo. Soy un verdadero experto en John Harris y sus problemas emocionales. No me gustaría basar mi plan de juego de 1983 y hacer pedidos de cartuchos ROM por 300.000 dólares basados en un juego que se supone que John Harris debe entregar. Si a su novia no le gustara, o dijera que es malo en la cama, se iría”.

“Si puedes hacer [Frogger] con el talento tonto que tenemos, imagina lo que sucederá cuando tengamos una empresa real. Seremos imparables. Si sigo dependiendo de tipos que podrían dejarme en cualquier momento porque alguien me ofrece más, o podrían dejar de trabajar de repente un día porque sus novias están saliendo con otra persona, entonces la empresa está condenada al fracaso en última instancia. Es solo cuestión de tiempo. Tengo que deshacerme de los llorones”.

Para Ken, el software, la magia, lo mesiánico, la transfiguración, la herramienta de la nueva era, había llegado a eso. Negocios. Separado de sus propias raíces de hacker, ya no parecía entender que los hackers no tomaban decisiones basadas en términos comerciales tradicionales, que algunos hackers no considerarían trabajar para empresas en las que no se sintieran cómodos, que algunos hackers eran reacios a trabajar para empresas en absoluto.

Pero a Ken no le importaba mucho lo que pensaran los hackers. Porque ya había terminado con ellos. Ken buscaba programadores profesionales, el tipo de personas orientadas a objetivos que abordaban una tarea como ingenieros responsables, no artistas prima donna obsesionados con hacer las cosas perfectas e impresionar a sus amigos. "Chicos buenos y sólidos que cumplan", fue la forma en que Ken lo expresó. "Perderemos nuestra dependencia de los programadores. Es tonto pensar que los programadores son creativos. En lugar de esperar a que llegue el correo, para que tipos como John Harris diseñen algo, vamos a conseguir algunos implementadores muy buenos que no son creativos, pero sí buenos".

Ken sintió que ya había encontrado algunos magos del juego latentes que habían estado enterrados en trabajos de programación corporativa. Uno de estos profesionales orientados a objetivos que Ken reclutó era un programador local de la compañía telefónica. Otro era un hombre de familia del sur de California de unos cuarenta años que había trabajado durante años en contratos gubernamentales utilizando imágenes digitales, según dijo, "con obvias implicaciones militares". Otro era un vegetariano rural de Idaho que vivía con su familia en una cúpula geodésica de madera.

Ken siguió adelante, tratando de reemplazar a los hackers por profesionales. Ya consideraba que el gran experimento que se estaba llevando a cabo en la vieja oficina de la Ruta 41, donde intentaba convertir a los novatos en programadores de lenguaje ensamblador, era un fracaso total. Llevaba demasiado tiempo formar a la gente y realmente no había nadie cerca que tuviera tanto el tiempo como el virtuosismo técnico para ser un gurú. Encontrar suficientes programadores de lenguaje ensamblador era difícil, e incluso una redada de cazatalentos y anuncios clasificados no podían garantizar los ganadores que Ken necesitaba el año siguiente. Necesitaría muchos, ya que su plan de juego de 1983 era lanzar más de cien productos. Pocos implicarían esfuerzos creativos originales. En cambio, la energía de programación de On-Line se concentraría en convertir sus juegos actuales a otras máquinas, especialmente a las computadoras de bajo costo, de mercado masivo y basadas en cartuchos ROM, como la VIC-20 o Texas Instruments. Las expectativas de On-Line se enunciaron en su “esquema de estrategia”: “Creemos que el mercado de las computadoras domésticas es tan explosivo que la ‘saturación de títulos’ es imposible. La cantidad de nuevas máquinas que compiten por el segmento Apple/Atari en 1983 creará un mercado perpetuamente nuevo, hambriento de los títulos ganadores de 1982. Aprovecharemos esta oportunidad…”.

La energía de la compañía se concentró en convertir un producto en otro producto. Fue un enfoque que desplazó la alegría de los hackers de crear nuevos mundos. En lugar de basarse en éxitos pasados en una búsqueda de programas brillantes, On-Line estaba tratando de maximizar las ventas duplicando incluso éxitos moderados, a menudo en máquinas relativamente limitadas en las que los juegos se veían peor que los originales. En medio de la vorágine de conversiones, no se previó recompensar un esfuerzo como el Frogger de Harris, que estaba tan logrado artísticamente que llegó al mercado con la fuerza de una obra original.

En su descuidado escritorio electrónico, John Harris filosofaba que los programadores "profesionales" (cualquier programador que no tuviera amor por los juegos en el corazón y perfeccionismo hacker en el alma) estaban destinados a crear juegos sin alma e imperfectos. Pero Ken Williams no estaba hablando con John Harris, quien, después de todo, ahora estaba programando para Synapse. Ken Williams estaba a punto de celebrar una reunión que pondría a On-Line en contacto con una nueva empresa, una que proporcionaría una línea de montaje completa de programadores profesionales para realizar conversiones. ¡A precios muy baratos!

Parecía demasiado bueno para ser verdad, y Ken entró en la reunión con sospecha. Su contacto en esta nueva empresa era un hombre de negocios de pelo largo hasta los hombros y ojos como los de Peter Lorre llamado Barry Friedman. La fortuna de Friedman había aumentado junto con la marea locamente creciente de la industria de las computadoras domésticas. En un principio, había representado a artistas que hacían ilustraciones para los anuncios y los embalajes de los productos de On-Line, y después se diversificó para acabar gestionándose de todo el material gráfico para unas cuantas empresas informáticas. A partir de ahí, empezó a prestar servicios a empresas de software con todo tipo de necesidades. Si querías saber dónde encontrar el mejor precio para cartuchos ROM, él podía actuar como intermediario para conseguirte ROM baratas, tal vez de algún oscuro proveedor de Hong Kong.

Últimamente, había estado insinuando que podía acceder a enormes sumas de capital a quienes lo necesitaran. El otro día, dijo Ken, Barry le había llamado y le había preguntado cuánto necesitaría un extraño para comprar On-Line. Ken sacó de la manga una cifra de 20 millones de dólares y colgó. Barry le devolvió la llamada ese mismo día diciendo que 20 millones de dólares estaba bien. Ken, que seguía sin tomárselo demasiado en serio, dijo: "Bueno, yo también necesitaría el control". Barry le devolvió la llamada poco después diciendo que eso también estaba bien. Lo curioso del asunto era que, por más escéptico que fuera Ken sobre Barry Friedman y su creciente grupo de empresas (nunca se podía estar seguro de qué nombre corporativo aparecería en la tarjeta de presentación que Barry o sus colegas te entregaban), siempre parecía cumplir sus promesas. Era como si Barry Friedman fuera el beneficiario de algún trato fáustico, al estilo de Silicon Valley.

Este nuevo trato sonaba como el más sorprendente de todos. Friedman acompañaba a la reunión con Ken Williams a los dos fundadores de una empresa emergente a la que representaba. Una empresa que no hacía nada más que conversiones. Las tarifas parecían de ganga: una tarifa de diez mil dólares y una regalía del cinco por ciento. La empresa se llamaba “Rich and Rich Synergistic Enterprises”, siendo Rich el primer nombre de ambos fundadores.

Barry Friedman, que llevaba un polo amarillo, desabrochado para revelar una cadena de oro que complementaba su pulsera de plata y diamantes y su reloj de oro, guió a Riches y a uno de sus socios, un hombre bajo, rubio y de nariz respingada vestido con un traje un tanto punk. Se trataba de Tracy Coats, un ex manager de música rock que representaba a patrocinadores de “una familia muy rica”. Esta información se transmitió en voz baja, con un gesto de complicidad de las cejas.

Sin más fanfarrias, se sentaron alrededor de la larga mesa de conferencias de madera en la sala de juntas que estaba junto a la oficina de Ken: una habitación perfectamente anodina, alfombrada y de paredes blancas con estanterías de madera y una pizarra; una habitación aleatoria y anónima que podría existir en cualquier pequeño complejo de oficinas de cualquier tipo de empresa.

“Rich y Rich…” dijo Ken, mirando los currículos de los dos programadores. “Espero que me hagáis rico”.

Ninguno de los dos se rió, y si sus rostros sin arrugas eran una indicación, la risa no era algo a lo que los Rich se permitieran en exceso. Todos eran gente de negocios y sus currículos eran incluso más serios que su apariencia. Ambos habían ocupado puestos de responsabilidad en el área de uso intensivo de la tecnología digital del recientemente terminado Disneyland Tokio (“Todo el lugar está basado en silicio”, dijo Rich One), pero esa fábrica de diversión autoritaria era lo más cercano a la frivolidad en sus currículos, que estaban repletos de frases como análisis de circuitos de serpientes, Laboratorio de Propulsión a Chorro, control nuclear, analista de sistemas de misiles,

sistema de control y guía interna de vuelo de misiles Hound Dog. Ambos Rich llevaban chaquetas deportivas sin corbatas y la ropa tenía ese aire de cuidado que adquiere la ropa cuando se coloca sobre cuerpos que se mantienen en forma compulsiva. Ambos parecían tener treinta y tantos años, con el pelo bien cortado y ojos atentos, constantemente escudriñando la habitación en busca de indiscreciones.

Rich Dos habló. “Nuestra gente proviene de un ámbito más profesional que otros en el campo de la informática doméstica. Gente que ha estado en un entorno más controlado que los tipos de ordenadores domésticos. Gente que sabe cómo documentar y escribir código correctamente”. Rich Dos hizo una pausa. “No son tipos hackers”, añadió.

Su empresa desarrollaría un conjunto de herramientas y técnicas para la conversión de juegos. Las técnicas, algoritmos y ensambladores cruzados serían, por supuesto, propietarios. Por eso, Rich and Rich mantendrían rutinariamente su código fuente. Lo guardarían en las oficinas de Rich and Rich en el sur de California. No importaba lo brillantes que fueran los trucos, no importaba lo elegante que fuera el trasero, no estaría disponible para el placer de la lectura de los hackers. Solo estaría disponible el producto. Opacidad. La gente compraría programas como productos, con la programación profundamente oculta, tan poco importante como la maquinaria que hace surcos en los discos que reproducen música. Del mismo modo, los programadores de Rich and Rich serían anónimos. No habría egos de hackers con los que lidiar. Solo tenían que enviar una lista de deseos de juegos y la cadena de montaje los produciría en serie.

A Ken le encantó la idea. “Los hará ricos y yo ganaré dinero”, dijo después. Si los dos proyectos de prueba que les dio a Rich y Rich funcionaban, dijo, “¡podría hacer todas mis conversiones con ellos! ¡Esto es mucho mejor que John Harris!”.

Ken se sentía en la cima de su juego. Además de Rich y Rich, un reportero del Wall Street Journal estaba en la ciudad, hablando con él y Roberta para un artículo sobre la empresa. Como solía hacer a mitad del día, se recompensó saliendo de la oficina y dirigiéndose al lugar donde se construiría su nueva casa. Ese día, estaban bajando las siete vigas del techo de siete metros de largo que irían sobre la enorme sala de juegos de la casa, no lejos de la cancha de racquetball cubierta. Se puso una camisa de franela sobre su raída camiseta azul de Apple y condujo hasta el lugar embarrado y observó cómo la grúa hidráulica levantaba las vigas y cómo el equipo de trabajo de doce hombres colocaba cada una en su nicho. Todo salió bien, como una subrutina bien escrita que funcionó la primera vez que el código se compilaba, y Ken miró con un orgullo aturdido lo que estaba construyendo. "¿No es extraño?", seguía preguntando. "¿No es extraño?".

La casa seguía y seguía, bajando por la colina durante cuarenta metros; El marco finalmente se llenó, con escaleras que se podían subir y puertas por las que se podía espiar. En ese momento, la casa estaba expuesta a los elementos, al viento y a la lluvia, y no había puertas ni paredes que impidieran el libre movimiento. Una casa hacker perfecta e infinita. Pero los constructores pronto colocarían paredes para evitar que el mundo mirara dentro de la casa y puertas para evitar que la gente de la casa entrara a la fuerza y violara la privacidad de una persona. Nadie en su sano juicio querría que fuera diferente.

Lo mismo con el hackerismo, tal vez... nadie que dirigiera una empresa podría querer que realmente la dirigiera la Ética Hacker. Tarde o temprano, tenías que enfrentarte a la realidad; anhelabas esas paredes y puertas antiguas y familiares que siempre se consideraron tan naturales que solo los locos las eliminarían. Tal vez solo en una simulación por computadora, usando la computadora para

hackear Utopia, podrías preservar ese tipo de idealismo. Tal vez ese era el único lugar donde podías preservar un sueño. En una computadora.

Ken caminó alrededor de la casa unas cuantas veces, habló con el constructor y luego recordó que tenía que regresar. Tuvo que hablar con el periodista del Wall Street Journal sobre la pequeña y extraña empresa familiar de software que había empezado con un juego de aventuras.

• • • • • • • •

Ken y Roberta Williams celebraron la fiesta de inauguración de la casa el fin de semana del Día del Trabajo de 1983. Más de doscientas personas pasearon por la casa de madera de cedro de diez mil pies cuadrados, admiraron las vidrieras, se maravillaron ante la chimenea de piedra de río, participaron en un torneo en la cancha de ráquetbol (que tenía un logotipo de Apple Computer a todo color incrustado en la madera reluciente), sudaron en la sauna, se relajaron en el jacuzzi, jugaron al tira y afloja en el patio trasero junto al río Fresno, remataron pelotas de vóleibol en la cancha, vieron videos transmitidos desde la antena parabólica del exterior, se rieron con la compañía de comediantes traída en avión desde San Francisco y jugaron a los seis juegos de arcade con monedas en la gigantesca sala de juegos con una barra de tamaño completo.

Fue una ocasión agridulce. Entre la competencia de los nuevos ricos, la crisis económica, el enorme desembolso de capital para cartuchos ROM que encajaran en máquinas de gama baja como la VIC-20 (gastos que nunca se recuperarían) y la falta de un nuevo éxito innovador programado por los hackers de Tercera Generación en Sierra On-Line, la empresa se encaminaba a un año con ingresos inferiores a los del año anterior. Ken se había visto obligado a buscar más capital de riesgo, tres millones de dólares de los cuales medio millón había ido directamente a parar a sus manos, considerablemente menos que el coste de la nueva casa.

A principios de ese verano, Ken le había pedido a Dick Sunderland que se reuniera con él en Broken Bit. Antes de que intercambiaran una palabra, Ken le entregó a su antiguo jefe una nota que decía: "Quedas despedido como presidente de Sierra On-Line". Dick Sunderland estaba furioso y acabó presentando una demanda contra Ken y On-Line. "Estoy furioso", explicaba. "Tengo mi reputación. Le he construido una empresa que puede dirigirse y él quiere dirigirla". Otros trabajadores de On-Line, especialmente aquellos que recordaban con cariño los días del campamento de verano, se alegraron. Sacaron la placa con el nombre de Sunderland de su plaza de aparcamiento y la pegaron en la puerta del baño de mujeres. Cogieron una pila de memorandos que databan del régimen de Sunderland, que se denominó “La era de la opresión”, y los arrojaron a una hoguera improvisada. Por un momento fugaz, fue como si los empleados de una empresa pudieran reducir la burocracia a cenizas.

Hubo otras notas optimistas. Ken tenía la esperanza de que su nuevo programa de procesamiento de textos de bajo coste le reportara dinero y que le iría bien un acuerdo de un millón de dólares para obtener la licencia de los personajes de dibujos animados de B.C. y The Wizard of Id. Estaba negociando con John Travolta el uso del nombre del actor en un programa de acondicionamiento físico. Pero a pesar de estos proyectos, el negocio del software había resultado ser más precario de lo que parecía en un principio.

Sólo había que hablar con Jerry Jewell para averiguar por qué: Jewell, de Sirius, había venido desde Sacramento y estaba lamentando el desastroso final de su contrato con Twentieth Century Fox Games: los juegos en cartucho que su compañía había escrito se habían perdido en la superabundancia de videojuegos de 1983 y él no había recibido casi nada de dinero a cambio de

centrar toda su ofensiva de mercado en la máquina Atari VCS. Su compañía pendía de un hilo y él dudaba de que alguno de los miembros de la Hermandad fuera capaz de sobrevivir en los próximos años. Sus mejores programadores lo habían dejado días antes de que él estuviera a punto de despedirlos.

Ken Williams también seguía teniendo problemas con los programadores. Estaba el hacker que dirigía el proyecto de IBM, muy retrasado respecto de lo previsto. Había algunos de los programadores "profesionales" que, al no estar familiarizados con los placeres de la inmersión en el universo de los juegos de ordenador, eran incapaces de sintetizar esos placeres por sí mismos. Incluso hubo una disputa con Bob y Carolyn Box: los dos buscadores de oro convertidos en programadores habían rechazado las críticas de Ken sobre el juego que le mostraron y habían dejado la empresa para convertirse en autores de software independientes.

Y luego estaba John Harris. Últimamente, él y Ken habían estado peleándose por un desacuerdo sobre regalías con el Frogger, todavía el programa más vendido de On-Line. Parker Brothers quería comprar el programa para convertirlo en cartucho, y Ken le ofreció a John el veinte por ciento de la compra de doscientos mil dólares. Para John eso no fue suficiente. Lo discutieron en la oficina de Ken. La discusión terminó con Ken Williams mirando a su ex superestrella del software y diciéndole: "Sal de mi oficina, John Harris. Me estás haciendo perder el tiempo".

Esa fue la última vez que hablaron antes de la inauguración de la casa, a la que Ken no había invitado a John. No obstante, Harris había aparecido con su novia, que lucía un gran anillo de compromiso de diamantes que él le había regalado. Ken saludó cordialmente al hacker. No era un día para animosidades, era un día para celebrar. Ken y Roberta Williams tenían su nueva casa de ochocientos mil dólares, y al menos no había nubes oscuras sobre las Sierras. La computadora les había proporcionado a todos riquezas y fama con las que nunca se habían atrevido a soñar, y mientras el anochecer se asomaba sobre el monte Deadwood, Ken Williams, vestido con pantalones cortos y una camiseta, bailaba alegremente al son de una banda de bluegrass que había traído del sur de California. Más tarde, tal como siempre soñaba, se sentó en el jacuzzi con amigos, un millonario de veintitantos años con un jacuzzi en las montañas. Mientras los amigos estaban sentados en el jacuzzi, con los brazos alrededor del borde, podían escuchar los débiles sonidos electrónicos de los juegos de arcade en la sala de juegos cercana, mezclándose incongruentemente con el susurro del bosque de Sierra.

Parte IV. El último de los Hackers Verdaderos: Cambridge: 1983

Apéndice A. El último de los verdaderos hackers

En la época de la fiesta de inauguración de la casa de Ken Williams, veinticinco años después de que el Club de Ferrocarriles Modelo del MIT Tech descubriera el TX-0, un hombre que se autodenominaba el último hacker verdadero estaba sentado en una habitación del noveno piso de Tech Square, una habitación abarrotada de impresiones, manuales, un saco de dormir y una terminal de ordenador parpadeante conectada a un descendiente directo del PDP-6, un ordenador DEC-20. Su nombre era Richard Stallman y hablaba con una voz tensa y aguda que no intentaba ocultar la emoción con la que describía, en sus palabras, “la violación del laboratorio de inteligencia artificial”. Tenía treinta años. Su tez pálida y su pelo oscuro y desaliñado contrastaban vívidamente con la intensa luminiscencia de sus profundos ojos verdes. Los ojos se humedecieron mientras describía la decadencia de la ética hacker en Tech Square.

Richard Stallman había llegado al MIT doce años antes, en 1971, y había experimentado la epifanía que otros habían disfrutado al descubrir ese paraíso puro de los hackers, el monasterio de Tech Square donde uno vivía para hackear y hackeaba para vivir. Stallman había estado fascinado con las computadoras desde la escuela secundaria. En un campamento de verano, se había divertido con manuales de computadoras que había tomado prestados de sus consejeros. En su Manhattan natal, encontró un centro de computación para ejercitar su nueva pasión. Cuando ingresó en Harvard, era un experto en lenguajes ensambladores, sistemas operativos y editores de texto. También había descubierto que tenía una profunda afinidad con la ética hacker y era militante en la ejecución de sus principios. Fue la búsqueda de una atmósfera más compatible con el hacking lo que lo llevó desde el relativamente autoritario centro de computación de Harvard, en la Avenida Massachusetts, al MIT.

Lo que le gustaba del laboratorio de IA de Tech Square era que “no había obstáculos artificiales, cosas que se imponen y que dificultan que la gente haga cualquier trabajo, como la burocracia, la seguridad, las negativas a compartir con otras personas”. También le encantaba estar con gente para la que el hacking era una forma de vida. Reconocía que su personalidad era inflexible ante el toma y daca de la interacción humana común. En el noveno piso se le podía apreciar por su hacking y formar parte de una comunidad construida en torno a esa búsqueda mágica.

Su magia pronto se hizo evidente, y Russ Noftsker, el administrador del laboratorio de IA que había tomado las duras medidas de seguridad durante las protestas de Vietnam, contrató a Stallman como programador de sistemas. Richard a menudo estaba en fase nocturna, y cuando la gente del laboratorio descubrió después del hecho que al mismo tiempo estaba obteniendo un título magna cum laude en física en Harvard, incluso esos maestros hackers se quedaron atónitos.

Cuando se sentó a los pies de figuras como Richard Greenblatt y Bill Gosper, a quienes consideraba su mentor, la visión de Stallman de la ética hacker se solidificó. Llegó a ver el laboratorio como la encarnación de esa filosofía; un anarquismo constructivo que, como escribió Stallman en un archivo de computadora una vez, “no significa defender una jungla de perros que se comen entre sí. La sociedad estadounidense ya es una jungla de perros que se comen entre sí, y sus reglas la mantienen así. Nosotros [los hackers] deseamos reemplazar esas reglas con una preocupación por la cooperación constructiva”.

Stallman, a quien le gustaba que lo llamaran por sus iniciales, RMS, en homenaje a la forma en que se conectaba a la computadora, utilizó la ética hacker como principio rector para su obra más

conocida, un programa de edición llamado EMACS que permitía a los usuarios personalizarlo sin límites; su arquitectura abierta alienta a las personas a agregarle cosas y mejorarlo sin cesar. Distribuyó el programa gratuitamente a todo aquel que aceptara su única condición: “que devolvieran todas las extensiones que hicieran, para ayudar a mejorar EMACS. Llamé a este acuerdo ‘la comunidad EMACS’”, escribió RMS. “Como yo compartía, era su deber compartir; trabajar con los demás en lugar de hacerlo en contra”.

EMACS se convirtió casi en un editor de texto estándar en los departamentos de informática de las universidades. Fue un ejemplo brillante de lo que podía producir el hacking.

Pero a medida que avanzaban los años setenta, Richard Stallman comenzó a ver cambios en su amado coto. La primera incursión fue cuando se asignaron contraseñas a usuarios oficialmente sancionados y se mantuvo fuera del sistema a los usuarios no autorizados. Como verdadero hacker, RMS despreciaba las contraseñas y estaba orgulloso del hecho de que las computadoras que le pagaban por mantener no las usaban. Pero el departamento de informática del MIT (dirigido por personas diferentes a las del laboratorio de IA) decidió instalar seguridad en su máquina.

Stallman hizo campaña para eliminar la práctica. Animó a la gente a utilizar la contraseña de “cadena vacía”, un retorno de carro en lugar de una palabra. Entonces, cuando la máquina te pedía tu contraseña, pulsabas la tecla RETORNO y te conectabas. Stallman también descifró el código de cifrado de la computadora y pudo acceder al archivo protegido que contenía las contraseñas de las personas. Comenzó a enviarles mensajes que aparecían en la pantalla cuando iniciaban sesión en el sistema:

> Veo que elegiste la contraseña [tal y tal]. Te sugiero que cambies a la contraseña a “Retorno de Carro.” [N.d.T. “Intro”] Es mucho más fácil de escribir y también cumple con el principio de que no debería haber contraseñas.

“Finalmente llegué a un punto en el que una quinta parte de todos los usuarios de la máquina tenían la cadena de contraseña vacía”, se jactó más tarde RMS.

Luego, el laboratorio de informática instaló un sistema de contraseñas más sofisticado en su otra computadora. Esta no fue tan fácil de descifrar para Stallman. Pero Stallman pudo estudiar el programa de cifrado y, como dijo más tarde, “descubrí que cambiar una palabra en ese programa haría que imprimiera su contraseña en la consola del sistema como parte del mensaje que indicaba que estaba iniciando sesión”. Dado que la “consola del sistema” era visible para cualquiera que pasara por allí y sus mensajes podían accederse fácilmente desde cualquier terminal, o incluso imprimirse en papel, el cambio de Stallman permitió que cualquier contraseña fuera difundida rutinariamente por cualquiera que quisiera saberla. Pensó que el resultado era “divertido”.

Aun así, el gigante de las contraseñas siguió adelante. El mundo exterior, con su afición por la seguridad y la burocracia, se estaba cerrando. La manía de la seguridad incluso infectó al sagrado ordenador de IA. El Departamento de Defensa amenazaba con sacar la máquina de IA de la red ARPAnet (para separar a la gente del MIT de la activa comunidad electrónica de hackers, usuarios y simples científicos informáticos de todo el país), todo porque el laboratorio de IA se negaba rotundamente a limitar el acceso a sus ordenadores. Los burócratas del Departamento de Defensa estaban furiosos: cualquiera podía entrar desde la calle y utilizar la máquina de IA, ¡y conectarse a otras ubicaciones de la red del Departamento de Defensa! Stallman y otros pensaban que así debía ser, pero él llegó a comprender que el número de personas que lo apoyaban estaba disminuyendo.

Cada vez más hackers incondicionales abandonaban el MIT, y muchos de los hackers que habían formado la cultura y le habían dado una columna vertebral con su comportamiento habían desaparecido hacía tiempo.

¿Qué había pasado con los hackers de antaño? Muchos se habían ido a trabajar para empresas, aceptando implícitamente los compromisos que ese trabajo implicaba. Peter Samson, el hacker de TMRC que fue uno de los primeros en descubrir el TX-0, estaba en San Francisco, todavía con la compañía Systems Concepts cofundada por el maestro hacker de teléfonos Stew Nelson. Samson podía explicar lo que había sucedido: "[El hackeo] ahora compite por la atención de uno con responsabilidades reales: trabajar para ganarse la vida, casarse, tener un hijo. Lo que tenía entonces y no tengo ahora es tiempo y una cierta cantidad de resistencia física". Era una conclusión común, más o menos compartida por personas como el colega de Samson en TMRC, Bob Saunders (que trabajaba para Hewlett-Packard, dos hijos en la escuela secundaria), David Silver (después de crecer en el laboratorio de inteligencia artificial, ahora dirigía una pequeña empresa de robótica en Cambridge), Slug Russell (el autor de Spacewar estaba programando para una empresa fuera de Boston y jugaba con su computadora personal Radio Shack), e incluso Stew Nelson, quien a pesar de permanecer en modo soltero se quejó de que en 1983 no era capaz de hackear tanto como le hubiera gustado. "Hoy en día todo es trabajo y no tenemos tanto tiempo para las cosas técnicas que nos gustaría hacer", dijo el hombre que hace más de dos décadas había utilizado instintivamente el PDP-1 para explorar el universo que era el sistema telefónico.

Nunca habría otra generación como ellos; Stallman se daba cuenta de esto cada vez que veía el comportamiento de los nuevos "turistas" aprovechando la libertad de la computadora de IA. No parecían tan bien intencionados ni tan ansiosos de sumergirse en la cultura como sus predecesores. En tiempos anteriores, la gente parecía reconocer que el sistema abierto era una invitación a hacer un buen trabajo y mejorarse a sí mismo hasta el punto de que un día podría ser considerado un verdadero hacker. Ahora, algunos de estos nuevos usuarios no podían soportar la libertad de curiosear en un sistema con los archivos de todos abiertos para ellos. "El mundo exterior está presionando", admitió Stallman. "Cada vez llega más gente que ha utilizado otros sistemas informáticos. En otros lugares, se da por sentado que si alguien puede modificar tus archivos, no podrás hacer nada, serás saboteado cada cinco minutos. Cada vez hay menos gente que haya crecido aquí a la antigua usanza y que sepa que es posible y que es una forma razonable de vivir".

Stallman siguió luchando, intentando, según dijo, "retrasar los avances fascistas con todos los métodos que pude". Aunque sus funciones oficiales de programación de sistemas se dividían por igual entre el departamento de informática y el laboratorio de inteligencia artificial, se puso en "huelga" contra el Laboratorio de Informática debido a su política de seguridad. Cuando sacó una nueva versión de su editor EMACS, se negó a dejar que el laboratorio de informática lo utilizara. Se dio cuenta de que, en cierto sentido, estaba castigando a los usuarios de esa máquina en lugar de a las personas que establecían la política. "Pero, ¿qué podía hacer?", dijo más tarde. "Las personas que usaban esa máquina aceptaban la política. No estaban luchando. Mucha gente estaba enfadada conmigo, diciendo que estaba intentando mantenerlos como rehenes o chantajearlos, lo que en cierto sentido era lo que estaba haciendo. Estaba participando en actos de violencia en contra de ellos porque pensaba que estaban cometiendo actos de violencia contra todo el mundo en general".

Las contraseñas no eran el único problema al que Richard Stallman tuvo que enfrentarse en lo que se estaba convirtiendo cada vez más en una defensa solitaria de la ética hacker pura en el MIT. Muchas de las nuevas personas que se encontraban en el laboratorio habían aprendido informática

en máquinas pequeñas y no estaban instruidas en los principios de los hackers. Al igual que los hackers de tercera generación, no veían nada malo en el concepto de propiedad de los programas. Estas nuevas personas escribirían programas nuevos y emocionantes, tal como lo hicieron sus predecesores, pero algo nuevo vendría con ellos: cuando los programas aparecieran en la pantalla, también lo harían los avisos de copyright. ¡Avisos de copyright! Para RMS, que todavía creía que toda la información debería fluir libremente, esto era una blasfemia. “No creo que el software deba ser propiedad de alguien”, dijo en 1983, años demasiado tarde. “Porque [la práctica] sabotea a la humanidad en su conjunto. Impide que las personas obtengan el máximo beneficio de la existencia del programa”.

Fue este tipo de comercialismo, en opinión de Richard Stallman, lo que asestó el golpe fatal a lo que quedaba de la comunidad idealista que había amado. Era una situación que encarnaba el mal y sumergía a los hackers restantes en un conflicto amargo. Todo comenzó con la máquina LISP de Greenblatt.

• • • • • • • •

Con el paso de los años, Richard Greenblatt había seguido siendo quizás el vínculo principal con los días de gloria de los hackers del noveno piso. A sus treinta y tantos años, el hacker resuelto de la Máquina de Ajedrez y MacLISP estaba moderando algunos de sus hábitos personales más extremos, arreglando su pelo corto con más frecuencia, variando más su vestuario e incluso pensando tentativamente en el sexo opuesto. Pero todavía podía hackear como un demonio. Y ahora estaba empezando a ver la realización de un sueño que había tenido hacía mucho tiempo: un ordenador totalmente hacker.

Se había dado cuenta de que el lenguaje LISP era extensible y lo suficientemente potente como para dar a la gente el control para construir y explorar el tipo de sistemas que podrían satisfacer la mentalidad de hacker más hambriento. El problema era que ningún ordenador podía manejar fácilmente las considerables demandas que LISP imponía a una máquina. Así que a principios de los años setenta Greenblatt comenzó a diseñar un ordenador que ejecutara LISP más rápido y con mayor eficiencia que cualquier máquina anterior. Sería una máquina para un solo usuario: finalmente una solución al problema estético del tiempo compartido, donde el hacker se siente psicológicamente frustrado por la falta de control definitivo sobre la máquina. Al ejecutar LISP, el lenguaje de la inteligencia artificial, la máquina sería un caballo de batalla pionero de la próxima generación de ordenadores, máquinas con la capacidad de aprender; de mantener diálogos inteligentes con el usuario sobre todo, desde el diseño de circuitos hasta las matemáticas avanzadas.

Así que, con una pequeña subvención, él y otros hackers (en particular Tom Knight, que había sido fundamental en el diseño (y el nombre) del Sistema de tiempo compartido incompatible) comenzaron a trabajar. Fue un proceso lento, pero en 1975 tenían lo que llamaban una máquina “Cons” (llamada así por la complicada función de “operador constructor” que la máquina realizaba en LISP). La máquina Cons no era independiente y tenía que estar conectada al PDP-10 para funcionar. Tenía dos bahías de ancho, con las placas de circuitos y la maraña de cables expuestas, y la construyeron allí mismo, en el noveno piso de Tech Square, en el piso elevado con aire acondicionado debajo.

Funcionó como Greenblatt esperaba. “LISP es un lenguaje muy fácil de implementar”, explicó Greenblatt más tarde. “Cada vez que un hacker se va a una máquina y trabaja duro durante un par de semanas y escribe un LISP. ‘Mira, tengo LISP’. Pero hay una gran diferencia entre eso y un

sistema realmente utilizable". La máquina Cons - y más tarde la máquina LISP autónoma - era un sistema utilizable. Tenía algo llamado "espacio de direcciones virtuales", que aseguraba que los programas de espacio consumidos no sobrecargaran rutinariamente la máquina, como era el caso en otros sistemas LISP. El mundo que se construía con LISP podía ser mucho más intrincado. Un hacker que trabajara en la máquina sería como un piloto de cohete mental viajando en un universo LISP en constante expansión.

Durante los siguientes años trabajaron para conseguir que la máquina fuera autónoma. El MIT pagaba sus salarios y, por supuesto, todos ellos también realizaban trabajos de sistemas en ITS y hacking aleatoria de IA. El cambio llegó cuando ARPA aportó dinero para que el grupo construyera seis máquinas por unos cincuenta mil dólares cada una. Después llegó otro dinero para construir más máquinas.

Al final, los hackers del MIT construirían treinta y dos máquinas LISP. Desde fuera, la computadora LISP parecía una unidad central de aire acondicionado. Toda la acción visual se desarrollaba en una terminal remota, con un teclado largo y elegante repleto de teclas de función y una pantalla de mapa de bits de altísima resolución. En el MIT, la idea era conectar varias máquinas LISP en una red, de modo que, si bien cada usuario tenía el control total, también podía hackear como parte de una comunidad, y los valores que surgían del flujo libre de la información pudiesen conservarse.

La máquina LISP era un logro significativo, pero Greenblatt se dio cuenta de que sería necesario algo más que fabricar unas cuantas máquinas y hackearlas. Esta máquina LISP era un creador de mundos flexible, la encarnación del sueño hacker... pero sus virtudes como "máquina pensante" también la convirtieron en una herramienta para que Estados Unidos mantuviera su liderazgo tecnológico en la carrera de la inteligencia artificial contra los japoneses. La máquina LISP tenía implicaciones más grandes que el laboratorio de IA, sin duda, y una tecnología como esta se difundiría mejor a través del sector comercial. Greenblatt: "En general, durante todo este proceso me di cuenta de que probablemente [íbamos] a fundar una empresa algún día y, finalmente, fabricaríamos estas máquinas LISP comercialmente. [Era] algo del tipo de cosas que tarde o temprano sucedería. Así que, a medida que la máquina se fue completando, empezamos a investigar".

Así fue como Russell Noftsker se metió en la situación. El ex administrador del laboratorio de IA había dejado su puesto bajo presión en 1973 y se había ido a California para empezar un negocio. De vez en cuando volvía a Cambridge y pasaba por el laboratorio para ver qué estaban haciendo los trabajadores de IA. Le gustaba la idea de las máquinas LISP y manifestó su interés en ayudar a los hackers a formar una empresa.

"Al principio, casi todo el mundo estaba en su contra", recordó Greenblatt más tarde. "Cuando Noftsker dejó el laboratorio, yo me llevaba mucho mejor con él que con cualquier otra persona. La mayoría de la gente odiaba a este tipo. Había hecho un montón de cosas que eran realmente muy paranoicas. Pero yo dije: 'Bueno, démosle una oportunidad'".

La gente lo odiaba, pero pronto quedó claro que Noftsker y Greenblatt tenían ideas diferentes de lo que debería ser una empresa. Greenblatt era demasiado hacker como para aceptar una estructura empresarial tradicional. Lo que quería era algo "que se acercara al patrón de la IA". No quería un montón de capital de riesgo. Greenblatt prefería un enfoque de autofinanciación, en el que la empresa recibiría un pedido de una máquina, la construiría, se quedaría con un porcentaje del dinero y lo reinvertiría en la empresa. Esperaba que su firma pudiera mantener un vínculo estable con el

MIT; incluso imaginó una manera en la que todos pudieran seguir afiliados al laboratorio de IA. El propio Greenblatt se resistía a irse; había establecido firmemente los parámetros de su universo. Si bien su imaginación tenía rienda suelta dentro de una computadora, su mundo físico todavía estaba en gran parte limitado por su desordenada oficina con terminal en el noveno piso y la habitación que había alquilado desde mediados de los años sesenta a un dentista jubilado (ahora fallecido) y a la esposa del dentista. Viajaría por todo el mundo para asistir a conferencias sobre inteligencia artificial, pero las discusiones en esos lugares remotos serían continuaciones de los mismos problemas técnicos que debatía en el laboratorio o en el correo informático de ARPAnet. La comunidad hacker lo definía mucho y, aunque sabía que la comercialización era necesaria hasta cierto punto para difundir el evangelio de la máquina LISP, quería evitar cualquier compromiso innecesario con la ética hacker: como las líneas de código en un programa de sistemas, el compromiso debe reducirse al mínimo.

Noftsker consideró que esto era poco realista y su punto de vista se filtró a los demás hackers involucrados en el proyecto. Además de Tom Knight, entre ellos había algunos jóvenes magos que no habían estado en la época dorada del noveno piso y tenían un enfoque más pragmático de lo que se requería. “Mi percepción [de la idea de Greenblatt] era comenzar una empresa que fabricara máquinas LISP en una especie de taller de garaje. Estaba claro que no era práctico”, dijo Tom Knight más tarde. “El mundo simplemente no es así. Solo hay una forma en que una empresa funciona y es tener gente motivada para ganar dinero”.

Knight y los demás percibieron que el modelo de empresa de Greenblatt era algo así como Systems Concepts en San Francisco, que incluía a los antiguos hackers del MIT Stewart Nelson y Peter Samson. Systems Concepts era una empresa de pequeña escala, guiada por una firme resolución de no tener que rendir cuentas a nadie que tuviera los hilos del dinero. “Nuestro objetivo inicial no era necesariamente hacernos infinitamente ricos”, explicó el cofundador Mike Levitt en 1983, “sino controlar nuestro propio destino. No le debemos nada a nadie”. Sin embargo, los hackers del MIT preguntaron cuál había sido el impacto de Systems Concepts; después de más de una década, concluyeron, todavía era pequeña y no muy influyente. Knight analizó Systems Concepts: “Bajo riesgo, no acepta financiación externa, no contrata a nadie que no conozca, ese modo”, dijo. “No va muy lejos”. Él y los demás tenían una visión más amplia para una empresa de máquinas LISP.

Russ Noftsker también vio, y explotó, el hecho de que muchos de los hackers eran reacios a trabajar en una empresa dirigida por Greenblatt. Greenblatt estaba tan concentrado en fabricar máquinas LISP, en la misión de hackear, en el trabajo que había que hacer, que a menudo se olvidaba de reconocer la humanidad de las personas. Y a medida que los hackers de antaño se hacían mayores, esto se convirtió en un problema cada vez mayor. “Todos lo toleraban por su brillantez y su capacidad de producción”, explicaría luego Nofisker, “[Pero] finalmente comenzó a usar sumisión o látigo para intentar que la gente hiciera lo que él quería. Reprendía a la gente que no estaba acostumbrada a ello. Los trataba como si fueran una especie de equipo de mulas de producción. Finalmente llegó al punto en que las comunicaciones se rompieron e incluso tomaron la medida extrema de mudarse del noveno piso para alejarse de Richard”.

Las cosas llegaron a un punto crítico en una reunión en febrero de 1979, cuando quedó claro que Greenblatt quería una empresa de estilo hacker y poder para asegurar que siguiera siendo así. Era una exigencia extraña, ya que durante mucho tiempo el laboratorio había, como dijo Knight, “gestionado sobre principios anarquistas, basados en el ideal de confianza mutua y respeto mutuo por la confianza técnica de las personas involucradas construida a lo largo de muchos años”. Pero el

anarquismo no parecía ser Lo Correcto en este caso. Tampoco lo era, para muchos, la exigencia de Greenblatt. “Francamente, no podía imaginarme que él cumpliera un rol presidencial en una compañía en la que yo estaba involucrado”, dijo Knight.

Noftsker: “Todos estábamos tratando de disuadirlo. Le rogamos que aceptara una estructura en la que él fuera igual al resto de nosotros y donde tuviéramos una gestión profesional. Y él se negó a hacerlo. Así que recorrimos la sala y preguntamos a cada persona del grupo técnico si aceptarían una organización que tuviera alguno de los elementos [que Greenblatt quería]. Y todos dijeron que no participarían en [tal] aventura”.

Fue un empate. La mayoría de los hackers no se unieron a Greenblatt, el padre de la máquina LISP. Noftsker y el resto dijeron que le darían a Greenblatt un año para formar su propia compañía, pero en algo menos de un año concluyeron que Greenblatt y los patrocinadores que logró encontrar para su LISP Machine Incorporated (LMI) no estaban “ganando”, por lo que formaron una compañía fuertemente capitalizada llamada Symbolics. Lamentaban tener que fabricar y vender las máquinas a las que Greenblatt había contribuido tanto, pero sentían que era necesario hacerlo. La gente de LMI se sentía traicionada; cada vez que Greenblatt hablaba de la división, su discurso se arrastraba hasta convertirse en un lento murmullo y buscaba formas de cambiar de tema. El amargo cisma era el tipo de cosas que podían suceder en los negocios o cuando la gente invertía emoción en las relaciones y la interacción humana, pero no era el tipo de cosas que se veían en la vida de los hackers.

El laboratorio de IA se convirtió en un campo de batalla virtual entre dos bandos, y las dos empresas, especialmente Symbolics, contrataron a muchos de los hackers que quedaban en el laboratorio. Incluso Bill Gosper, que había estado trabajando en Stanford y Xerox durante ese tiempo, acabó uniéndose al nuevo centro de investigación que Symbolics había formado en Palo Alto. Cuando Symbolics se quejó del posible conflicto de intereses de la gente de LMI que trabajaba para el laboratorio de IA (consideraba que el MIT, al pagar salarios a esos trabajadores a tiempo parcial de LMI, estaba financiando a su competencia), los hackers que todavía estaban afiliados al laboratorio, incluido Greenblatt, tuvieron que dimitir.

Fue doloroso para todos, y cuando ambas empresas lanzaron versiones similares de máquinas LISP a principios de los años 80, quedó claro que el problema iba a durar mucho tiempo. Greenblatt había hecho algunas concesiones en su plan de negocios (por ejemplo, llegó a un acuerdo por el cual LMI obtenía dinero y apoyo de Texas Instruments a cambio de una cuarta parte de las acciones) y su empresa estaba sobreviviendo. La más lujosa Symbolics había contratado a la flor y nata del hackerismo e incluso había firmado un contrato para vender sus máquinas al MIT. Lo peor fue que la comunidad ideal de hackers, esas personas que, en palabras de Ed Fredkin, “se amaban entre sí”, ya no se hablaban. “Me gustaría mucho hablar con [Greenblatt]”, dijo Gosper, hablando en nombre de muchos hackers de Symbolics que prácticamente habían crecido con los hackers más canónicos y ahora estaban aislados de su flujo de información. “No sé hasta qué punto está contento o descontento conmigo por haberme unido a los malos aquí. Pero lo siento, me temo que esta vez tenían razón”.

Pero incluso si las personas de las empresas hablaban entre sí, no podían hablar de lo que más importaba: la magia que habían descubierto y forjado dentro de los sistemas informáticos. La magia era ahora un secreto comercial, no para que la examinaran las empresas competidoras. Al trabajar

para empresas, los miembros de la sociedad hacker purista habían descartado el elemento clave de la ética hacker: el libre flujo de información. El mundo exterior estaba dentro.

• • • • • • • •

La persona que más se vio afectada por el cisma y su efecto en el laboratorio de IA fue Richard Stallman. Se lamentaba de que el laboratorio no defendiera la ética hacker. RMS les decía a los extraños que conocía que su esposa había muerto, y no sería hasta más adelante en la conversación que el extraño se daría cuenta de que ese joven delgado y quejoso estaba hablando de una institución en lugar de una novia perdida trágicamente.

Más tarde, Stallman escribió sus pensamientos en la computadora:

> Es doloroso para mí traer de vuelta los recuerdos de esta época. Las personas que permanecen en el laboratorio eran los profesores, estudiantes e investigadores no hackers que no sabían cómo mantener el sistema o el hardware, o querían saberlo. Las máquinas empezaron a romperse y nunca se arreglaban; a veces simplemente las tiraban a la basura. No se podían hacer los cambios necesarios en el software. Los no hackers reaccionaron a esto recurriendo a sistemas comerciales, trayendo consigo el fascismo y los acuerdos de licencia. Yo solía pasear por el laboratorio, por las salas tan vacías por la noche que antes estaban llenas y pensaba: "¡Oh, mi pobre laboratorio de IA! Te estás muriendo y no puedo salvarte". Todos esperaban que si se capacitaban más hackers, Symbolics los contrataría, así que ni siquiera parecía que valiera la pena intentarlo... toda la cultura fue aniquilada...

Stallman lamentaba el hecho de que ya no fuera fácil pasarse por allí o llamar a la hora de la cena y encontrar un grupo ansioso por una cena china. Llamaba al número del laboratorio, que terminaba en 6765 (“Fibonacci de 20”, solía decir la gente, señalando un rasgo numérico establecido al principio por algún hacker matemático cualquiera), y no encontraba a nadie con quien comer ni con quién hablar.

Richard Stallman sintió que había identificado al villano que destruyó el laboratorio: Symbolics. Hizo un juramento: “Nunca usaré una máquina Symbolic LISP ni ayudaré a nadie a hacerlo... No quiero hablar con nadie que trabaje para Symbolics ni con la gente que trata con ellos”. Si bien también desaprobaba la empresa LMI de Greenblatt, porque como empresa vendía programas informáticos que Stallman creía que el mundo debería tener gratis, sentía que LMI había intentado evitar dañar al laboratorio de IA. Pero Symbolics, en opinión de Stallman, había despojado deliberadamente al laboratorio de sus hackers para evitar que donaran tecnología de la competencia al dominio público.

Stallman quería contraatacar. Su campo de batalla era el sistema operativo LISP, que originalmente era compartido por el MIT, LMI y Symbolics. Esto cambió cuando Symbolics decidió que los frutos de su trabajo serían de su propiedad; ¿por qué LMI debería beneficiarse de las mejoras realizadas por los hackers de Symbolics? Así que no habría intercambio. En lugar de que dos empresas unieran sus esfuerzos para lograr un sistema operativo con muchas funciones, tendrían que trabajar de forma independiente, gastando energía en duplicar las mejoras.

Esta era la oportunidad de venganza de RMS. Dejó de lado sus escrúpulos sobre LMI y comenzó a cooperar con esa empresa. Como todavía estaba oficialmente en el MIT y Symbolics instalaba sus mejoras en las máquinas del MIT, Stallman pudo reconstruir cuidadosamente cada nueva característica o corrección de un error. Luego reflexionaba sobre cómo se había realizado el cambio,

lo comparaba y presentaba su trabajo a LMI. No era un trabajo fácil, ya que no podía simplemente duplicar los cambios: tenía que encontrar formas innovadoras y diferentes de implementarlos. "No creo que haya nada inmoral en copiar código", explicó. “Pero demandarían a LMI si copiara su código, por lo tanto tengo que hacer mucho trabajo”. RMS, un auténtico John Henry del código informático, había intentado por sí solo igualar el trabajo de más de una docena de hackers de talla mundial, y se las había arreglado para seguir haciéndolo durante la mayor parte de 1982 y casi todo 1983. “En un sentido bastante real”, señaló Greenblatt en ese momento, “ha estado superando a todos ellos en hackeo”.

Algunos hackers de Symbolics se quejaron no tanto por lo que estaba haciendo Stallman, sino porque no estaban de acuerdo con algunas de las decisiones técnicas que Stallman tomó en la implementación. “Realmente me pregunto si esa gente no se está engañando a sí misma”, dijo Bill Gosper, él mismo dividido entre la lealtad a Symbolics y la admiración por el hack maestro de Stallman. “O si están siendo justos. “Puedo ver algo que escribió Stallman, y podría decidir que era malo (probablemente no, pero alguien podría convencerme de que era malo), y aún así diría: ‘Pero espera un minuto, Stallman no tiene a nadie con quien discutirlo toda la noche. ¡Está trabajando solo! ¡Es increíble que alguien pueda hacer esto solo!’”

Russ Noftsker, presidente de Symbolics, no compartía la admiración de Greenblatt o Gosper. Se sentaba en las oficinas de Symbolics, relativamente lujosas y bien decoradas en comparación con la destartalada sede de LMI a una milla de distancia, su rostro infantil se arrugaba de preocupación cuando hablaba de Stallman. “Desarrollamos un programa o una mejora para nuestro sistema operativo y lo hacemos funcionar, y eso puede llevar tres meses, y luego, según nuestro acuerdo con el MIT, se lo entregamos. Y luego [Stallman] lo compara con los antiguos y lo analiza y ve cómo funciona y lo reimplementa [para las máquinas LMI]. Lo llama ingeniería inversa. Lo llamamos robo de secretos comerciales. No tiene ningún sentido en el MIT que lo haga porque ya le hemos dado esa función [al MIT]. El único propósito que tiene es dársela a la gente de Greenblatt”.

Y ese era exactamente el objetivo. Stallman no se hacía ilusiones de que su acto mejoraría significativamente el mundo en general. Había llegado a aceptar que el entorno del laboratorio de IA había sido contaminado de forma permanente. Estaba dispuesto a causarle al culpable el mayor daño posible. Sabía que no podía seguir así indefinidamente. Fijó una fecha límite para su trabajo: finales de 1983. Después de eso no estaba seguro de cuál sería su próximo paso.

Se consideraba el último hacker auténtico que quedaba en la Tierra. “El laboratorio de IA solía ser el único ejemplo que demostraba que era posible tener una institución anarquista y muy buena”, explicaba. “Si le dijera a la gente que es posible no tener seguridad en un ordenador sin que la gente borre tus archivos todo el tiempo y sin jefes que te impidan hacer cosas, al menos podría señalar el laboratorio de IA y decir: ‘Miren, lo estamos haciendo. ¡Vengan a usar nuestra máquina! ¡Vean!’ Ya no puedo hacer eso. Sin este ejemplo, nadie me creerá. Durante un tiempo estuvimos dando ejemplo al resto del mundo. Ahora que esto ha desaparecido, ¿por dónde voy a empezar? El otro día leí un libro. Se llama Ishi, el último Yahi. “Es un libro sobre el último sobreviviente de una tribu de indios, inicialmente con su familia, y luego gradualmente se fueron extinguiendo uno por uno”.

Así se sentía Richard Stallman. Como Ishi.

“Soy el último sobreviviente de una cultura muerta”, dijo RMS. “Y realmente ya no pertenezco a este mundo. Y en cierto modo siento que debería estar muerto”.

Richard Stallman dejó el MIT, pero se fue con un plan: escribir una versión del popular sistema operativo de computadoras propietario llamado UNIX y dárselo a cualquiera que lo quisiera. Trabajar en este programa GNU (que significaba “Gnu no es Unix”) significaba que podía “seguir usando computadoras sin violar [sus] principios”. Al ver que la ética hacker no podía sobrevivir en la forma pura en la que había prosperado anteriormente en el MIT, se dio cuenta de que numerosos actos pequeños como el suyo mantendrían viva la ética en el mundo exterior.

• • • • • • •

Lo que hizo Stallman fue unirse a un movimiento masivo de hackerismo en el mundo real que se puso en marcha en la misma institución que estaba abandonando con tanto dolor. El surgimiento del hackerismo en el MIT veinticinco años antes fue un intento concentrado de ingerir por completo la magia de la computadora; de absorber, explorar y expandir las complejidades de esos sistemas fascinantes; de usar esos sistemas perfectamente lógicos como inspiración para una cultura y una forma de vida. Fueron estos objetivos los que motivaron el comportamiento de Lee Felsenstein y los hackers de hardware desde Albuquerque hasta el Área de la Bahía. El feliz resultado de sus acciones fue la industria de las computadoras personales, que expuso la magia a millones de personas. Sólo un porcentaje minúsculo de estos nuevos usuarios de computadoras experimentarían esa magia con la furia omnipresente de los hackers del MIT, pero todos tuvieron la oportunidad de hacerlo... y muchos verían destellos de las posibilidades milagrosas de la máquina. Extendería sus poderes, estimularía su creatividad y les enseñaría algo, tal vez, de la ética hacker, si escuchaban.

A medida que la revolución informática crecía en una vertiginosa espiral ascendente de silicio, dinero, publicidad exagerada e idealismo, la ética hacker se volvió tal vez menos pura, un resultado inevitable de su conflicto con los valores del mundo exterior. Pero sus ideas se extendieron por toda la cultura cada vez que algún usuario encendía la máquina y la pantalla cobraba vida con palabras, pensamientos, imágenes y, a veces, mundos elaborados construidos a partir del aire: esos programas informáticos que podían convertir a cualquier hombre (o mujer) en un dios.

A veces, los pioneros más puros se quedaban asombrados por su progenie. Por ejemplo, Bill Gosper se sorprendió con un encuentro en la primavera de 1983. Aunque Gosper trabajaba para la empresa Symbolics y se dio cuenta de que, en cierto sentido, se había vendido al hackear en el sector comercial, seguía siendo el mismo Bill Gosper que una vez se sentó en el PDP-6 del noveno piso como un alquimista gregario del código. Se lo podía encontrar a altas horas de la madrugada en una habitación del segundo piso cerca de El Camino Real en Palo Alto, su destartalado Volvo era el único coche en el pequeño aparcamiento fuera del anodino edificio de dos plantas que albergaba el centro de investigación de Symbolics en la Costa Oeste. Gosper, que ahora tenía cuarenta años, sus rasgos afilados ocultos tras unas grandes gafas de montura metálica y el pelo recogido en una coleta que le llegaba hasta la mitad de la espalda, seguía hackeando LIFE, observando con desenfrenada diversión cómo la terminal de su máquina LISP procesaba miles de millones de generaciones de colonias LIFE.

“Tuve la experiencia más increíble cuando fui a ver El retorno del Jedi”, dijo Gosper. “Me senté junto a un chico de quince o dieciséis años. Le pregunté qué hacía y me dijo: ‘Oh, básicamente soy un hacker’. Casi me caigo. No dije nada. No estaba preparado para eso. Sonó como la cosa más arrogante que he oído nunca”.

El joven no estaba alardeando, por supuesto, sino describiendo quién era. Un hacker de Tercera Generación. Con muchas más generaciones por seguir.

Para los pioneros como Lee Felsenstein, esa continuidad representaba un objetivo cumplido. El diseñador del Sol y del Osborne 1, el cofundador de Community Memory, el héroe de la novela pseudo-Heinlein de su propia imaginación, a menudo se jactaba de haber estado “presente en la creación” y de haber visto los efectos del boom que siguió a una distancia lo suficientemente cercana como para ver sus limitaciones y su influencia sutil, significativa. Después de que hizo su fortuna en papel en Osborne, la vio desvanecerse con la misma rapidez, ya que la mala gestión y las ideas arrogantes sobre el mercado hicieron que Osborne Computer colapsara en un período de unos pocos meses en 1983. Se negó a lamentar su pérdida financiera. En cambio, se enorgullecía de celebrar que "el mito de la megamáquina más grande que todos nosotros [el malvado gigante descomunal, accesible solo por el sacerdocio] ha sido enterrado. Podemos volver a descender de la adoración de la máquina".

Lee Felsenstein había aprendido a usar un traje con facilidad, a cortejar a las mujeres, a encantar al público. Pero lo que importaba seguía siendo la máquina y su impacto en las personas. Tenía planes para el siguiente paso. "Hay más por hacer", dijo poco después de que Osborne Computer se hundiera. "Tenemos que encontrar una relación entre el hombre y la máquina que sea mucho más simbiótica. Una cosa es descender de un mito, pero hay que reemplazarlo por otro. Creo que hay que empezar por la herramienta: la herramienta es la encarnación del mito. Estoy intentando ver cómo se puede explicar el futuro de esa manera, crear el futuro”.

Se sentía orgulloso de haber ganado su primera batalla (llevar las computadoras a la gente). Incluso mientras hablaba, la Tercera Generación de hackers estaba haciendo noticia, no sólo como diseñadores de juegos superestrella, sino como tipos de héroes culturales que desafiaban los límites y exploraban los sistemas informáticos. Una película de gran éxito llamada WarGames tenía como protagonista a un hacker de Tercera Generación que, sin tener conocimiento de las hazañas revolucionarias de Stew Nelson o Captain Crunch, irrumpía en los sistemas informáticos con el asombro inocente de su Imperativo de Manos a la Obra. Era un ejemplo más de cómo la computadora podía difundir la Ética.

“La tecnología tiene que ser considerada como algo más que las piezas inanimadas de hardware”, dijo Felsenstein. “La tecnología representa formas inanimadas de pensar, formas objetivadas de pensar. El mito que vemos en WarGames y cosas así es definitivamente el triunfo del individuo sobre el desánimo colectivo. [El mito] intenta decir que la sabiduría convencional y los entendimientos comunes siempre deben estar abiertos a cuestionamiento. No es sólo un punto académico. Es un punto muy fundamental, se podría decir, de la supervivencia de la humanidad, en el sentido de que se puede tener gente [simplemente] que sobreviva, pero la humanidad es algo un poco más precioso, un poco más frágil. De modo que poder desafiar una cultura que dice “No tocarás esto” y desafiar eso con los propios poderes creativos es… la esencia”.

La esencia, por supuesto, de la ética hacker.

Apéndice B. Epílogo: Diez años después

> Creo que los hackers (programadores informáticos dedicados, innovadores e irreverentes) son el grupo de intelectuales más interesante y eficaz desde los redactores de la Constitución de los Estados Unidos... Ningún otro grupo que yo conozca se ha propuesto liberar una tecnología y lo ha logrado. No sólo lo hicieron en contra del desinterés activo de las corporaciones estadounidenses, sino que su éxito obligó a las corporaciones estadounidenses a adoptar su estilo al final. Al reorganizar la Era de la Información en torno al individuo, a través de las computadoras personales, los hackers bien pueden haber salvado la economía estadounidense... La más silenciosa de todas las subculturas de los años 60 ha surgido como la más innovadora y poderosa.
>
> --Stewart Brand Fundador, Whole Earth Catalog

En noviembre de 1984, en los húmedos y ventosos promontorios al norte de San Francisco, ciento cincuenta programadores canónicos y tecno-ninjas se reunieron para la primera Conferencia Hacker. Este evento, que en un principio fue ideado por Stewart Brand, fundador de Whole Earth Catalog, transformó un campamento militar abandonado en una sede mundial temporal de la ética hacker. No es casualidad que el evento coincidiera con la publicación de este libro, y un buen número de los personajes que aparecen en sus páginas se presentaron, en muchos casos para conocerse por primera vez. Los hackers del MIT de primera generación, como Richard Greenblatt, se juntaron con eminencias del Homebrew como Lee Felsenstein y Stephen Wozniak y los zares del juego Ken Williams, Jerry Jewell y Doug Carlston. Los magos impetuosos de la nueva computadora Macintosh se reunieron con gente que hackeaba Spacewar. Todos dormían en literas, lavaban platos y limpiaban mesas, y dormían mínimamente. Durante unas horas se fue la electricidad y la gente charlaba a la luz de las linternas. Cuando se restableció la electricidad, la avalancha hacia la sala de computadoras, donde uno podía exhibir sus hacks, fue algo que probablemente no se había visto en este país desde la última estampida de búfalos.

Recuerdo que pensé: “Estos son los verdaderos hackers”.

Me encontraba en un estado de gran ansiedad, sentado entre ciento cincuenta potenciales críticos que habían recibido copias de mi primer libro. Los incluidos en el texto inmediatamente buscaron sus nombres en el índice y procedieron a examinar los pasajes para comprobar su precisión y corrección tecnológica. Los que no estaban en el índice se enfadaron y hasta el día de hoy, cada vez que me encuentran, en persona o en el éter del ciberespacio, se quejan. En definitiva, la experiencia fue estimulante. La Conferencia Hacker, que se convertiría en un evento anual, resultó ser el puntapié inicial de un debate animado y público, que continúa hasta el día de hoy, sobre el futuro del hacking y la ética hacker tal como se define en este libro.

El término “hacker” siempre ha estado plagado de discusiones. Cuando estaba escribiendo este libro, el término todavía era bastante desconocido. De hecho, algunos meses antes de la publicación, mi editor me dijo que la gente del equipo de ventas de Doubleday pidió un cambio de título: “¿Quién sabe qué es un hacker?”, preguntaron. Afortunadamente, nos quedamos con el original y, a mediados de los años ochenta, el término ya se había arraigado en el lenguaje común.

Sin embargo, por desgracia para muchos verdaderos hackers, la popularización del término fue un desastre. ¿Por qué? La palabra hacker había adquirido una connotación específica y negativa. El

problema comenzó con algunas detenciones muy publicitadas de adolescentes que se aventuraron electrónicamente en terrenos digitales prohibidos, como los sistemas informáticos del gobierno. Era comprensible que los periodistas que cubrían estas historias se refirieran a los jóvenes delincuentes como hackers; después de todo, así es como se llamaban a sí mismos los jóvenes. Pero la palabra rápidamente se convirtió en sinónimo de "intruso digital".

En las páginas de las revistas nacionales, en los dramas y películas de televisión, en las novelas pulp y de prestigio, surgió un estereotipo: el hacker, un geek antisocial cuyo atributo identificativo es la capacidad de sentarse frente a un teclado y conjurar un tipo de magia criminal. En estas descripciones, cualquier cosa conectada a una máquina de cualquier tipo, desde un misil nuclear hasta una puerta de garaje, es fácilmente controlada por los huesudos dedos del hacker, que teclea en un PC barato o una estación de trabajo. Según esta definición, un hacker es, en el mejor de los casos, una persona benigna, un inocente que no se da cuenta de sus verdaderos poderes. En el peor, es un terrorista. En los últimos años, con la aparición de los virus informáticos, el hacker se ha transformado literalmente en una fuerza virulenta.

Es cierto que algunos de los hackers más honrados de la historia han sido conocidos por burlarse de detalles como los derechos de propiedad o el código legal para cumplir con el Imperativo de la Manos a la Obra. Y las bromas siempre han sido parte del hacking. Pero la inferencia de que esas travesuras eran la esencia del hacking no sólo era errónea, sino que resultaba ofensiva para los verdaderos hackers, cuyo trabajo había cambiado el mundo y cuyos métodos podían cambiar la forma en que uno veía el mundo. Leer sobre estudiantes de secundaria sin talento que se conectan a los tablones de anuncios de las computadoras, descargan contraseñas del sistema o códigos de agencias de crédito y los utilizan para promover el caos digital, y que los medios los llamen hackers... bueno, era demasiado para las personas que se consideraban los auténticas. Se pusieron furiosos. La comunidad hacker todavía hierve de rabia por la crítica pública que recibió en 1988 en la Hacker Conference 5.0, cuando un equipo de reporteros de CBS News se presentó aparentemente para hacer un reportaje sobre la gloria de los hackers canónicos, pero en su lugar publicó un artículo repleto de especialistas en seguridad que advertían sobre la *Amenaza de los Hackers*. Hasta el día de hoy, creo que Dan Rather haría bien en evitar asistir a futuras Conferencias Hacker.

Pero en los últimos años, creo que la marea ha cambiado. Cada vez más personas han aprendido sobre el espíritu del verdadero hacking tal como se describe en estas páginas. No sólo los que tienen conocimientos técnicos conocen las ideas e ideales de los hackers, sino que los aprecian y se dan cuenta, como insinuó Brand, de que son algo que hay que cultivar.

Varias cosas han contribuido a esta transformación. Primero fue la propia revolución informática. A medida que el número de personas que utilizan ordenadores aumentó de cientos de miles a cientos de millones, la magia proteica de la máquina difundió su mensaje implícito, y quienes se inclinaban a explorar sus poderes buscaron naturalmente a sus antecesores.

En segundo lugar, estaba la Red. Millones de personas están conectadas entre sí en redes informáticas, y la mayor parte de los hackers serios se unen a los diez millones de personas que forman parte de la confederación llamada Internet. Es un conducto que conecta a las personas entre sí, facilitando proyectos de colaboración. Y también es un semillero de conferencias y conversaciones, una cantidad sorprendente de las cuales tratan sobre cuestiones derivadas de la ética hacker y sus conflictos con las finanzas y el mundo real.

Por último, los verdaderos hackers se volvieron populares. Bajo el nombre de “ciberpunk”, un término apropiado de las novelas negras futuristas de nuevos escritores de ciencia ficción inteligentes como William Gibson, Bruce Sterling y Rudy Rucker, surgió un nuevo movimiento cultural a principios de los años 90. Cuando la publicación insignia del movimiento, Mondo 2000 (un cambio de nombre de Reality Hackers) comenzó a dilucidar los principios del ciberpunk, resultó que la mayoría de ellos se originaron en la ética hacker. Las creencias implícitas del Tech Model Railroad Club del MIT (La información debe ser libre, el acceso a las computadoras debe ser ilimitado y total, desconfiar de la autoridad...) han sido barajados y quedaron en la cima del mazo.

Cuando el ciberpunk llegó al espíritu de la época, los medios estaban listos para adoptar una visión más amplia y positiva del hacking. Había publicaciones enteras cuyo punto de vista era paralelo a los principios hacker: Mondo 2000 y Wired, y un montón de fanzines con nombres como Intertek y Boing Boing. Había una activa prensa comercial de computadoras escrita por periodistas que sabían que su industria debía su existencia a los hackers. Aún más significativo, los conceptos del hackerismo fueron adoptados por periodistas de las mismas publicaciones tradicionales cuya ignorancia había contaminado el hackerismo para empezar.

Una vez que la gente comprendió qué motivaba a los hackers, fue posible utilizar esas ideas como medida para examinar los valores de Silicon Valley. En Apple Computer en particular, los ideales de los hackers se consideraron cruciales para el bienestar de la empresa... su propia alma. Incluso las empresas más estrictas se dieron cuenta de que si querían ser líderes en sus campos, se necesitaba la energía, la visión y la perseverancia para resolver problemas de los hackers. A su vez, se exigiría a las empresas que relajaran sus reglas, para adaptarse al estilo libre de los hackers.

Lo mejor de todo es que estas ideas comenzaron a fluir más allá de la industria informática y a la cultura en general. Como aprendí mientras escribía Hackers, los ideales de mi tema podían aplicarse a casi cualquier actividad que uno persiguiera con pasión. Burrell Smith, el diseñador de la computadora Macintosh, lo dijo tan bien como cualquiera en una de las sesiones de la primera Conferencia Hacker: "Los hackers pueden hacer casi cualquier cosa y ser un hacker. Puedes ser un carpintero hacker. No es necesariamente alta tecnología. Creo que tiene que ver con la artesanía y el cuidado de lo que haces”.

Por último, una actualización de algunos personajes principales de Hackers, una década después.

Bill Gosper es un consultor que vive en Silicon Valley. Sigue hackeando, buscando los secretos de las matemáticas, los fractales y el juego LIFE, mientras se gana la vida como consultor. También sigue siendo soltero, y le explicó a un entrevistador en el libro More Mathematical People que tener hijos, o incluso una pareja, sería problemático porque “no importa cuán consciente sea el esfuerzo que haga para darles a los niños la atención que merecen, ellos sentirán que la computadora está ganando”.

La empresa LISP Machines de Richard Greenblatt fue devorada por las fauces corporativas. Después de trabajar como consultor, ahora dirige su propia pequeña empresa, dedicada a fabricar dispositivos médicos que combinan información de voz y datos a través de líneas telefónicas. Piensa mucho en el futuro del hacking y lamenta el día en que la comercialización abrumó el tipo de proyectos que se realizaban rutinariamente (con financiación del gobierno) en el MIT en los días dorados. Pero, dice, “la buena noticia es que el costo de este material está bajando tan rápidamente que es posible hacer cosas como un pasatiempo, entre comillas. Es posible hacer un trabajo serio por tu cuenta”.

• • • • • • • •

A diferencia de algunos de sus compañeros pioneros de las computadoras personales de la era del Homebrew, Lee Felsenstein nunca se hizo rico. Aunque disfrutó de fama dentro de la cultura tecnológica, sus propias empresas, dirigidas a través de su empresa Golemics en dificultades, siguieron siendo marginales. Sin embargo, recientemente consiguió un trabajo de ensueño como ingeniero líder en Interval, una nueva empresa de Silicon Valley bien financiada dedicada a inventar la próxima generación de magia técnica. A medida que se acerca a los cincuenta, la vida personal de Lee está más asentada: ha tenido varias relaciones serias y actualmente vive con una mujer que conoció a través de la red informática Whole Earth 'Lectronic Link. Sigue comprometido apasionadamente con el cambio social a través de las computadoras. Hace tiempo que hace circular la idea de formar una especie de Boy Scouts digitales (aunque sin género específico) llamada Hacker’s League. Y todavía cree que Community Memory, una vez que entre en la red, tendrá un impacto en el mundo.

• • • • • • • •

Ken Williams sigue siendo presidente de Sierra On-Line. La empresa ha tenido sus altibajos, pero al igual que su exitoso competidor Brøderbund y a diferencia de la extinta Sirius, es más grande que nunca, empleando a unas 700 personas en su sede de Oakhurst. Sierra salió a bolsa en 1992; las acciones de Ken lo convierten en millonario varias veces. Sierra también ha invertido millones de dólares en una red de juegos de computadora interactivos; AT&T ha comprado el veinte por ciento de la empresa. Roberta Williams es la diseñadora de juegos más popular de Sierra, aclamada por su serie King’s Quest de juegos de aventuras gráficas en 3D.

Ken Williams cree que en Sierra hay poco espacio para el antiguo espíritu hacker. “En los primeros tiempos, una persona, John Harris, podía hacer un proyecto”, dice Ken. “Ahora, nuestros juegos tienen cincuenta o más nombres en los créditos. No hacemos ningún producto sin un presupuesto de desarrollo de al menos un millón. En King’s Quest VI, hay un guión de setecientas páginas, leído por más de cincuenta actores profesionales. Fue el proyecto de grabación de voz más grande jamás realizado en Hollywood”.

Ken Williams me cuenta que John Harris todavía vive en la zona de Oakhurst y dirige una pequeña empresa que vende software para generar pantallas de visualización para operadores de televisión por cable. Según Williams, John Harris todavía está escribiendo su software para el ordenador Atari 800, que hace tiempo que se dejó de fabricar.

• • • • • • • •

Como se podría esperar del último hacker auténtico, Richard Stallman se ha mantenido fiel a los ideales del laboratorio de Inteligencia Artificial del MIT. Su fundación, la Free Software Foundation, es, según Wired, “la única organización benéfica del mundo con la misión de desarrollar software libre”. Stallman también ha sido una fuerza instrumental en la League for Software Freedom, un grupo que refleja su creencia de que el software propietario es una plaga para el paisaje digital. En 1991, sus esfuerzos llamaron la atención de los encargados de repartir las codiciadas becas McArthur Fellowship, “becas para genios”. La última vez que lo vi, Stallman estaba organizando una manifestación contra Lotus Development Corporation. Su protesta se refería a sus patentes de software. Creía, y todavía cree, que la información debería ser libre.

—Steven Levy, agosto de 1993

Apéndice C. Epílogo: 2010

“Es curioso”, dice Bill Gates. “Cuando era joven, no conocía a ninguna persona mayor. Cuando hicimos la revolución de los microprocesadores, no había nadie mayor, nadie. No nos hicieron reunirnos con periodistas que fueran personas mayores. No traté con personas de 30 años. Ahora hay personas de 50 y 60 años. Y ahora soy viejo y tengo que soportarlo. Es extraño lo vieja que se ha vuelto esta industria. Cuando era joven me reuní con ustedes, y ahora que soy viejo, me reúno con ustedes. ¡Jesús!”.

El cofundador de Microsoft y yo, un par de viejos de cincuenta y tantos años, estamos dando seguimiento a la entrevista que realicé para Hackers con un Gates despeinado hace más de un cuarto de siglo. Yo estaba tratando de capturar lo que yo pensaba que era el núcleo candente de la entonces floreciente revolución informática: los hackers, personas terriblemente obsesivas, absurdamente inteligentes e infinitamente inventivas. Gates apenas estaba empezando a cosechar los frutos de su acuerdo para suministrar su sistema operativo DOS a IBM, lo que posicionaría a Microsoft para dominar los PC de escritorio durante décadas. Su nombre todavía no era una palabra familiar. La palabra todavía no era una palabra familiar. Posteriormente entrevistaría a Gates muchas veces, pero esa primera entrevista fue especial. Vi su pasión por las computadoras como una cuestión de importancia histórica. Gates encontró mi interés en cosas como su “Carta a los aficionados” como una novedad intrigante. Pero para entonces yo estaba convencido de que mi proyecto era en verdad un registro de un movimiento que afectaría a todos.

Mi editor me había instado a ser ambicioso, y para mi primer libro, apunté alto, argumentando que los brillantes programadores que descubrieron mundos en la computadora fueron los actores clave en una transformación digital radical. Este enfoque de pensamiento a gran escala no era mi intención original. Cuando me embarqué en mi proyecto, pensaba que los hackers eran poco más que una subcultura interesante. Pero a medida que avanzaba mi investigación, descubrí que su carácter lúdico, así como su despreocupado desprecio por lo que otros decían que no se podía hacer, condujeron a los avances que determinaron el modo en que miles de millones de personas utilizaban las computadoras. Los hackers del MIT ayudaron a crear los videojuegos y los procesadores de textos. El Homebrew Computer Club alquimizó las matemáticas difíciles de la Ley de Moore en algo que terminó en todos nuestros escritorios, a pesar de la creencia predominante de que nadie necesitaría ni querría nunca una computadora personal. Y la mayoría de estos hackers lo hicieron simplemente por el placer de realizar un truco asombroso.

Detrás de la inventiva, descubrí algo aún más maravilloso: los hackers reales, sin importar cuándo o dónde surgieron, compartían un conjunto de valores que resultaron ser un credo para la era de la información. Intenté codificar este código tácito en una serie de principios que llamé La ética del hacker. Esperaba que estas ideas (en particular la creencia de los hackers de que “la información debería ser libre”) hicieran que la gente viera a los hackers desde una perspectiva diferente.

Aunque el libro inicialmente cayó con un golpe seco (el New York Times lo llamó “una historia de revista monstruosamente exagerada”), finalmente encontró su público, más allá incluso de mis expectativas exageradas. A través de encuentros casuales, correos electrónicos y tuits, la gente me dice constantemente que leer Hackers los inspiró en sus carreras o en su forma de pensar. Hojeando un libro sobre el creador de Doom, John Carmack, descubrí que leer Hackers le aseguraba al

adolescente geek que no estaba solo en el mundo. Cuando entrevisté recientemente a Ben Fried, el director de información de Google, apareció con una copia de Hackers con las esquinas dobladas para que se la firmara. “No estaría aquí hoy si no hubiera leído esto”, me dijo. Escucho eso docenas de veces al año y nunca me canso de eso.

Igualmente satisfactorio es el hecho de que las cuestiones planteadas en el libro se han convertido en algunas de las principales controversias de la era de la información. En la semana de la publicación del libro, muchos de mis sujetos (junto con otros hackers notables que no había incluido) se reunieron en el condado de Marin, California, para la primera Conferencia de Hackers. Fue allí donde Stewart Brand, el padrino de los hackers y editor de Whole Earth Catalog, hackeó el principio de que “la información debería ser gratuita”. Vale la pena citar su comentario, pronunciado de manera espontánea en una sesión que organicé llamada “El futuro de la ética hacker”, porque a menudo se lo cita incorrectamente. “Por un lado, la información quiere ser cara, porque es muy valiosa”, dijo Brand. “La información correcta en el lugar correcto simplemente cambia tu vida. Por otro lado, la información quiere ser gratuita, porque el costo de obtenerla es cada vez menor. Así que tienes a estos dos luchando entre sí”.

Un cuarto de siglo después, la reformulación de Brand es tan familiar que se ha convertido en un adjetivo (los críticos hablan de la “multitud de los que quieren que la información sea libre”). Pero la cita completa encapsula perfectamente la tensión que ha definido al movimiento hacker durante el último cuarto de siglo: una batalla a menudo acalorada entre el idealismo geek y el comercio despiadado. Los hackers quieren que la información sea libre, no necesariamente libre como la cerveza, sino libre como la libertad, para citar a Richard Stallman. Afortunadamente, el temor de Stallman de convertirse en Ishi, el último Yahi, no se hizo realidad.

El mundo de los hackers ha experimentado cambios radicales desde que escribí Hackers en un ordenador Apple II con WordStar (en los disquetes que se utilizaban en aquel entonces sólo podía caber la mitad de un capítulo). Casi nadie sabía qué era un hacker; algunos de los vendedores de la editorial original, Doubleday, insistieron en que se cambiara el título del libro por su falta de claridad. Internet era una red poco conocida que conectaba unos pocos ordenadores en el gobierno y en el mundo académico. Las personas que pasaban mucho tiempo con un ordenador eran consideradas antisociales y poco aptas para la conversación. Y algunas de las ideas que se esconden detrás del peculiar conjunto de valores de La ética del hacker ahora parecen tan obvias que los nuevos lectores pueden preguntarse por qué me molesté siquiera en escribirlas («¿Se puede crear arte y belleza en un ordenador?» ¿No?).

Ahora que Hackers se acerca a su 25º aniversario, me propuse volver a analizar el hackerismo, volviendo a visitar a algunas de las personas que conocí mientras investigaba para el libro. Mis visitas también incluyeron a algunos que no lo lograron la primera vez, principalmente porque aún no habían dejado su huella en el mundo hacker. Parte de mi búsqueda era ver qué significaba ser un hacker en 2010. Pero otra motivación era simplemente reconectarme con aquellos congelados en la extraña cesura que ocurre cuando los retratos quedan bloqueados en la impresión. Al igual que con el viaje por carretera de Bill Murray para visitar a antiguas novias en la película Flores rotas, esperaba extraer algún significado de ver lo que les había sucedido a mis sujetos, con la esperanza de que arrojaran luz sobre lo que ha sucedido con el hacking, y tal vez dieran una idea de cómo el hacking ha cambiado el mundo, y viceversa.

Solo pude visitar una pequeña muestra, pero en sus ejemplos encontré un reflejo de cómo se ha desarrollado el mundo tecnológico en los últimos veinticinco años. Si bien el movimiento hacker ha triunfado, no todas las personas que lo crearon corrieron la misma suerte. Como Gates, algunas de las personas en Hackers ahora son ricas, famosas y poderosas. Prosperaron en la transición del movimiento desde una subcultura insular a una industria multimillonaria, incluso si eso significaba de alguna manera desviarse del Camino Verdadero Hacker. Otros, que no querían o no podían adaptarse a un mundo que había descubierto y explotado su pasión (o simplemente tuvieron mala suerte), trabajaron en la oscuridad y lucharon por evitar la amargura. También vi el surgimiento de una nueva ola: los herederos actuales del legado hacker que crecieron en un mundo donde el comercio y los hackers nunca fueron vistos como valores opuestos. Están moldeando el futuro del movimiento.

• • • • • • •

Los verdaderos hackers no se toman vacaciones. Y a juzgar por esos estándares, Bill Gates ya no es un verdadero hacker.

El propio Gates lo admite. “Creo en la intensidad y estoy totalmente de acuerdo en que, según medidas objetivas, mi intensidad en la adolescencia y en los veinte años era más extrema”, afirma. “En los veinte, me limitaba a trabajar. Ahora voy a casa a cenar. Cuando decides casarte y tener hijos, si vas a hacerlo bien, vas a renunciar a parte del fanatismo”. De hecho, echando la vista atrás, Gates dice que los años clave de su etapa hacker llegaron incluso antes, cuando era adolescente en la Escuela Lakeside. “Los años más duros, los más fanáticos, van de los trece a los dieciséis”, dice.

“¿Así que ya estabas acabado cuando llegaste a Harvard?”, pregunto.

“¿En términos de programación veinticuatro horas al día? Ah, sí”, dice. “Sin duda, cuando tenía diecisiete años, mi mentalidad de software ya estaba formada”.

Me pregunto cómo un chico de hoy, cuando las computadoras son omnipresentes y fáciles de controlar, podría tener un impacto similar. ¿Podría haber un Bill Gates hoy? “Bueno, ciertamente no existe la oportunidad de llevar las computadoras a las masas”, dice. El gran estallido de la revolución informática ya se ha escuchado. Sin embargo, dice, “hay estallidos más grandes”. En algún lugar, cree Gates, podría haber algún genio que, a partir de un conjunto de hojas en blanco, creará una industria entera. Cuando sugiero que es difícil encontrar hojas en blanco, me ignora. “Hay toneladas”, dice. “En robótica. En inteligencia artificial. En programación de ADN. Y cinco o seis cosas que ni siquiera puedo nombrar porque no soy joven. Tenemos ciento treinta y cinco millones de personas que nacen cada año; no necesitamos un alto porcentaje. Ni siquiera necesitas uno al año. Así que puedes ser extremadamente exigente”.

Todavía parecía muy intenso cuando lo conocí a los veintisiete años: impetuoso pero reacio a hacer contacto visual directo. Durante la mitad de la entrevista, miró fijamente una pantalla de computadora, probando un software que usaba uno de esos ratones de última generación. Pero se involucró plenamente en mis preguntas, exponiendo su opinión muy firme sobre algunas de las personas con las que trabajó (y contra las que trabajó) en los primeros días de la PC. Esa intensidad influiría en su trabajo y en su empresa, ayudándolo a convertir a Microsoft en la principal empresa de software del mundo y a convertirlo, durante un tiempo, en el ser humano más rico del mundo. La fe de Gates en el hacking subyacía en todo su trabajo, incluso en sus decisiones sobre la contratación de personal. "Si quieres contratar a un ingeniero" - dice - “mira el código de ese tipo. Eso es todo. Si no ha escrito mucho código, no lo contrates”.

Repaso el incidente de su “Carta abierta a los aficionados” de 1976. “Lo planteé en el sentido de: ‘Dios, si la gente pagara más por el software, podría contratar a más gente’”, dice ahora.

¿Podría haber imaginado que esos problemas seguirían existiendo tantos años después? La respuesta fue sí, y su explicación es una mini lección de historia de la ley de propiedad intelectual, que se remonta a las teorías de Adam Smith y la reimpresión no autorizada y no pagada de los escritos de Benjamin Franklin por parte de editoriales europeas. “Benjamin Franklin fue tan estafado”, dice Gates. “Podría haber escrito exactamente lo que escribí. “¡Esa maldita imprenta!” Gates cree que nos espera un largo período de prueba de nuevos modelos de negocio para encontrar el equilibrio adecuado entre los titulares de los derechos y los lectores en la era digital. Y, al menos a mis oídos, parece albergar cierta satisfacción por el hecho de que ahora sean los periodistas los que se quejan de lo mismo por lo que él se quejaba en su carta. “Tal vez los escritores de revistas sigan cobrando dentro de veinte años”, dice. “O tal vez tengan que cortar el pelo durante el día y escribir artículos solo por la noche. ¿Quién sabe?”.

Gates tuvo que alejarse del rígido código moral de los hackers para convertirse en un éxito generalizado. Todo lo que Steve Wozniak tuvo que hacer fue ponerse un par de zapatos de baile. Si bien Woz sigue siendo una leyenda de los hackers, también se ha convertido en un improbable icono de la cultura pop, al aparecer en el exitoso programa Dancing with the Stars. Cuando me encontré con él para una nueva entrevista veinticinco años después, acababa de reunirse con otros concursantes para un final de temporada. "Estaba bailando contra Jerry Springer y Cloris Leachman", dice, mientras comía patatas fritas y salsa en un restaurante mexicano en Fremont, California. Su eliminación temprana de ninguna manera le empañó el ánimo. Muy pocas cosas empañan el ánimo de Woz, incluso el hecho de que la celebridad de los reality shows esté eclipsando sus logros genuinos en la historia de la tecnología. "La gente se me acerca y me dice: '¡Dios mío, te vi en Dancing with the Stars!' Tengo que decir: 'Bueno, también hice computadoras'".

Se puede perdonar a los fanáticos ocasionales por pasar por alto las credenciales tecnológicas de Woz. Hoy en día, es más probable que reciba atención por sus pasatiempos (como el polo en Segway) o por su vida amorosa: tuvo un romance apocalíptico con la comediante Kathy Griffin, aunque luego se casó con una mujer que conoció en un crucero Geek. Los sitios web sarcásticos se han burlado sin piedad de las apariciones de Woz en revistas de celebridades y sus frecuentes apariciones en las filas del primer día de la Apple Store como indicadores de una triste irrelevancia. Pero Woz se encoge de hombros alegremente. Recuerda la instrucción que le dio a Griffin hace unos años: "Oye, puedes avergonzarme, puedes insultarme, puedes ridiculizarme tanto como quieras; si hace reír a la gente, vale la pena". Cuando lo conocí a principios de los años 80, Woz era un millonario socialmente torpe y peligrosamente vulnerable. Ahora, es una figura paterna a prueba de balas y muy querida: una mascota para la cultura hacker en general.

De vez en cuando, Woz aparece en las noticias como impulsor de una startup con una tecnología potencialmente revolucionaria. CL 9 iba a idear controles remotos superpoderosos. Wheels of Zeus prometía permitir a los usuarios rastrear sus pertenencias mediante tecnología inalámbrica. Pero el primero no tuvo éxito y el segundo nunca salió al mercado. Ahora trabaja como científico jefe de una empresa de almacenamiento llamada Fusion-io. “Hablo en nombre del producto, hago mucho trabajo de marketing y ventas, pero también estoy estudiando tecnologías que podrían ser competitivas en el futuro”.

Pero ni siquiera Woz espera crear otro Apple II. En 2010, su mayor contribución es como modelo a seguir. Su renombre universal es un recordatorio constante de que el cerebro y la creatividad pueden triunfar sobre las nociones tradicionales de lo que es genial. Es el nerd de la sala de ordenadores cuya estatura (y felicidad) eclipsa con creces la de los reyes del baile de graduación caídos. Y eso es una inspiración para los nerds de todo el mundo.

De hecho, uno de sus protegidos, Andy Hertzfeld, sigue inspirado por el hacking. Hertzfeld no fue una figura importante en Hackers, pero como uno de los primeros empleados más brillantes de Apple Computer, podría haberlo sido. (Lo conocí a fines de 1983, cuando era uno de los diseñadores del sistema operativo Macintosh). Hoy está en Google, donde su contribución más visible es una función que crea cronologías para las consultas de Google News para que los usuarios puedan ver una historia en el contexto de su tiempo. Pero hackear a los cincuenta años no es tan fácil como hacerlo a los veinte. “Cuando hackeaba en Mac, estaba trabajando y pensaba que había pasado una hora. Luego miraba hacia arriba y habían pasado cuatro horas”, dice. “Ahora, cuando creo que ha pasado una hora, miro hacia arriba y es una hora”.

No es solo el paso de los años lo que ha cambiado la experiencia de Hertzfeld. También ha tenido que adaptar su enfoque individualista para servir al complejo industrial-geek que es Google. Por un lado, Google es la meca de los hackers. Valora a los ingenieros como su activo más importante. “Se espera que trabajes por tu pasión”, dice Hertzfeld, un valor definitivamente favorable para los hackers. Sin embargo, Hertzfeld no puede ignorar el hecho de que Google también es una gran empresa con estándares y procesos estrictos en lo que se refiere al diseño de productos, lo que hace que todo el proceso sea más formal y menos divertido. “Mi relación con mi trabajo es la de un artista con su trabajo”, dice. En Google, añade, “no puedo ejercitar mi creatividad de una manera que me dé alegría, que es mi enfoque básico”.

Pero si bien ha perdido algo de control personal, ha ganado una capacidad sin precedentes para dejar una marca en el mundo. Debido a la ubicuidad de las computadoras e Internet, con unas pocas líneas de código una persona en Google o Apple puede hacer un cambio que mejore la vida de millones de personas. Y eso genera un tipo de emoción diferente al que experimentó Hertzfeld durante los primeros días en Apple. “¿Sabes qué fue lo emocionante del Apple II?”, dice. “Podíamos hacer sonar el altavoz. Pero sabíamos que algún día podría hacer música. Por eso fue tan emocionante: cuando las cosas son más potenciales que realizaciones, esa es la máxima emoción. Por otro lado, ahora hay mucho más poder para generar un gran impacto. Estas cosas son tan comunes como pueden serlo hoy en día. Google, el iPhone... estos mueven la cultura más que los Beatles en los años sesenta. Están moldeando a la raza humana”.

• • • • • • • •

Richard Greenblatt me dice que tiene una perorata que soltar.

Uh-oh.

Después de todos estos años, ¿finalmente se va a quejar de la forma en que hablé de su higiene personal en los primeros capítulos de Hackers?

Para mi alivio, Greenblatt está más preocupado por lo que él ve como el estado decrépito de la informática. Odia los lenguajes de codificación dominantes de la actualidad como HTML y C++. Extraña LISP, el lenguaje amado con el que trabajó cuando estaba en el MIT. “El mundo está

jodido", dice, antes de lanzarse a un análisis técnico del estado actual de la programación que ni siquiera tengo la esperanza de seguir.

Pero la programación es sólo el principio. El verdadero problema, dice Greenblatt, es que los intereses comerciales han invadido una cultura que se construyó sobre los ideales de apertura y creatividad. En la época dorada de Greenblatt, él y sus amigos compartían código libremente, dedicándose exclusivamente al objetivo de crear mejores productos. "Ahora hay una dinámica que dice: 'Formatemos nuestra página web de modo que la gente tenga que pulsar el botón muchas veces para ver muchos anuncios'", dice Greenblatt. "Básicamente, la gente que gana es la que consigue hacer que las cosas sean lo más incómodas para ti".

Greenblatt no es una de esas personas. Pertenece a un grupo diferente: los verdaderos creyentes que todavía se aferran a su motivación original (la alegría del descubrimiento, el libre intercambio de ideas) incluso cuando su pasión brilla a la sombra de una industria multimillonaria. A pesar de su brillantez e importancia, nunca lanzaron un producto de un millón de dólares, nunca se convirtieron en un icono. Simplemente siguieron hackeando.

Estoy rodeado de idealistas similares aquí en la 25ª Conferencia de Hackers, que ha seguido siendo una reunión anual que celebra la emoción de construir algo realmente genial. Han pasado algunos años desde la última vez que asistí, pero es tal como lo recuerdo: cuarenta y ocho horas de hackers reunidos hasta bien entrada la noche en un resort de Santa Cruz, discutiendo sobre todo, desde teoría económica hasta almacenamiento masivo de datos. El público es algo mayor, a pesar de un esfuerzo tardío por atraer a más personas menores de treinta años. Los viejos siguen en pie.

Greenblatt es un habitual aquí, un vínculo con la Mesopotamia de la cultura hacker, el MIT. En estos días, Greenblatt se describe a sí mismo como un investigador independiente. Hace varios años se mudó a la casa de su madre en Cambridge para cuidarla en los últimos años de su vida y ha vivido allí solo desde que ella murió en 2005. Se mantiene en contacto con algunos de sus colegas del MIT y durante años ha intentado convencer al otro gran hacker canónico del Proyecto MAC, Bill Gosper, para que asista a una Conferencia de Hackers. Pero el brillante Gosper, algo ermitaño, nunca ha accedido. (Gosper, que también sigue hackeando, vive en Silicon Valley y vende acertijos matemáticos en su sitio web). "El proyecto principal en el que he estado trabajando durante quince años se llama memoria de hilos, y tiene algo que ver con la comprensión del idioma inglés", dice Greenblatt. "Es investigación básica. No es algo que funcione hoy, pero de todos modos, es algo".

Cuando Greenblatt observa el estado actual del hacking, ve un mundo en decadencia. Incluso la palabra en sí ha perdido su significado. Cuando le pregunto cuál es el estado del hacking hoy, su respuesta es instantánea y sincera. "Nos robaron la palabra", dice, "y se ha ido irremediablemente".

Greenblatt no es el único que invoca con nostalgia el pasado. Incluso cuando entrevisté por primera vez a Richard Stallman en 1983, se lamentaba del triste declive de la cultura hacker y pensaba que la comercialización del software era un crimen. Yo había asumido que el mundo pronto aplastaría a "El último de los verdaderos hackers" como a un insecto.

¡Qué equivocado estaba! La cruzada de Stallman por el software libre ha seguido informando las luchas en curso sobre la propiedad intelectual y Stallman obtuvo una beca de la Fundación MacArthur para "genios". Fundó la Free Software Foundation y escribió el sistema operativo GNU, que se adoptó ampliamente después de que Linus Torvalds escribiera Linux para ejecutarlo; la combinación se utiliza en millones de dispositivos. Tal vez lo más importante es que Stallman proporcionó el marco intelectual que llevó al movimiento de software libre, un elemento crítico del

software moderno y de la propia Internet. Si el software tuviera sus santos, Stallman habría sido beatificado hace mucho tiempo.

Sin embargo, es casi tan famoso por su personalidad inquebrantable. En 2002, el fundador de Creative Commons, Lawrence Lessig, escribió: “No conozco bien a Stallman. Lo conozco lo suficiente como para saber que es un hombre difícil de querer”. (¡Y eso estaba en el prefacio del propio libro de Stallman!). El tiempo no lo ha ablandado. En nuestra entrevista original, Stallman había dicho: “Soy el último sobreviviente de una cultura muerta. Y realmente ya no pertenezco al mundo. Y en cierto modo siento que debería estar muerto”. Ahora, mientras nos reunimos a comer comida china, por supuesto, lo reafirma: “Desde luego, me hubiera gustado suicidarme al nacer”, dice. “En términos de efectos sobre el mundo, es muy bueno que haya vivido. Así que supongo que, si pudiera volver atrás en el tiempo y evitar mi nacimiento, no lo haría. Pero me gustaría no haber sufrido tanto”.

El dolor se debía en parte a la soledad, que en su día era una queja común entre el pequeño y obsesivo grupo de aficionados a las computadoras (un comentario de 1980 del psicólogo de Stanford Philip Zimbardo daba a entender que los hackers eran perdedores antisociales que recurrían a las computadoras para evitar el contacto humano). Pero a medida que se difundía la cultura hacker, también lo hacía su aceptabilidad social. Hoy en día, los expertos en computadoras no son vistos como perdedores, sino como magnates en ciernes. Tienden a no sufrir el intenso aislamiento que ha plagado a Stallman, gracias, irónicamente, a la comercialización que tanto lamenta.

Hoy, como hace veinticinco años, Stallman es un fundamentalista, un huterita del hackerismo. Su sitio web personal es un revoltijo de llamamientos al boicot a varios enemigos de la causa, desde el Blu-Ray hasta J.K. Rowling. Incluso se pelea con sus antiguos aliados, incluido Torvalds. (“No quiere defender la libertad del usuario”, dice Stallman). Siente un desprecio particular por Apple, con sus sistemas cerrados y su software de derechos digitales. Se refiere a sus productos utilizando juegos de palabras al estilo de la revista Mad. El reproductor de música es un iScrod. Su dispositivo móvil es un iGroan. La nueva tableta es el iBad. Y él es un quejica que no tiene nada que envidiar a nadie. Cuando le digo que Hackers pronto estará disponible en un Kindle (al que Stallman, como era previsible, llama una estafa), su actitud adusta se evapora mientras me anima enérgicamente a resistirme al oneroso DRM del lector electrónico. “Hay que creer que la libertad es importante y que uno la merece”, dice. A pesar de su desilusión, el fuego sigue ardiendo en su interior.

Lee Felsenstein también mantiene viva la llama. De todas las personas sobre las que escribí en Hackers, Felsenstein fue el que habló más explícitamente de las consecuencias políticas de la revolución informática. Pero desde su triunfo con Osborne, su propia carrera ha estado llena de altibajos. Trabajó durante ocho años con el laboratorio de innovación Interval Research, pero ese esfuerzo fracasó. Varios otros proyectos parecían prometedores, incluido un esfuerzo para distribuir un servicio telefónico por Internet a Laos que iba a ser alimentado por generadores de bicicletas, pero por una razón u otra no despegó. “Si quisiera, podría estar amargado por ello, pero no quiero”, dice.

Aunque Felsenstein previó el auge de los ordenadores personales, todavía está esperando el tipo de democratización que esperaba que lo acompañara, cuando los ordenadores baratos en manos “del pueblo” permitirían a todo el mundo tomar información, manipularla para reflejar mejor la verdad y distribuirla ampliamente. “Está empezando a suceder, pero no de la forma en que yo había

supuesto”, afirma. “Lincoln Steffens comentó una vez: ‘He visto el futuro y funciona’, pero estoy de acuerdo con el tipo que lo cambió por: ‘He visto el futuro y necesita trabajo’”.

Felsenstein vio con consternación la erosión del término “hacker”, pero cree que ahora está en auge. “Hacker ahora tiene la connotación de alguien al límite y con más probabilidades de hacer el bien que el mal. Así que creo que estamos ganando la guerra cultural que todo el mundo creía que habíamos perdido en los años 80”. Por su parte, Felsenstein está poniendo a la próxima generación de geeks en el camino de la rectitud. Recientemente ayudó a establecer un espacio de trabajo en Mountain View, California, llamado Hacker Dojo, que cobra a sus ochenta miembros 100 dólares al mes por el acceso a un espacio de 9500 pies cuadrados con una red interna y herramientas extrañas como lectores IR. Es un puesto de avanzada en un número creciente de "Hacker Spaces" en todo el país dedicados a empoderar a los fanáticos de los equipos que antes estaban aislados y mal equipados. "Soy un sensei del dojo, que como sabrán es un gran maestro reverenciado", dice, con una amplia sonrisa en su rostro. "Sensei Felsenstein".

• • • • • • •

Greenblatt, Stallman y Felsenstein ven el hacking como un conjunto de ideales, pero Paul Graham lo ve como un motor económico en marcha. Este gurú de Internet de cuarenta y cinco años, que en su época fue un ingeniero fanático, es cofundador de Y Combinator, una incubadora de empresas emergentes de Internet. Dos veces al año, su empresa organiza concursos al estilo de American Idol para seleccionar de veinte a treinta empresas emergentes que participarán en un campamento de entrenamiento de tres meses, que culmina con un día de demostración repleto de inversores ángeles, inversores de capital riesgo y empresas ávidas de adquisiciones como Google y Yahoo.

¿Cómo elige Graham a los candidatos más prometedores? Fácil. Busca a los hackers. “Somos bastante hackers, así que es fácil reconocer a un espíritu afín”, dice Graham, que en 1995 co-creó Viaweb, la primera aplicación basada en la web. “Los hackers entienden un sistema lo suficientemente bien como para estar a cargo de él y hacer que cumpla sus órdenes, y tal vez hacer que haga cosas que no estaban previstas”. Los mejores prospectos, dice, son los “hackers mundiales”, gente “que no sólo sabe cómo jugar con las computadoras, sino que juega con todo”. De hecho, Graham dice que hoy en día, todas las compañías están buscando contratar o invertir en compañías dirigidas por hackers. “Les decimos a los fundadores que se presentan en el Demo Day: ‘Si te vistes demasiado elegante, los inversores te verán como una persona estúpida’. Vienen a ver al próximo Larry y Sergey, no a algún tipo con un MBA junior”.

Stallman se horrorizaría si Graham equiparara el hacking con la efectividad empresarial. Pero Graham ha descubierto que los valores del hacking no se ven amenazados por las empresas: han conquistado las empresas. Resolución de problemas improvisada. Toma de decisiones descentralizada. Poner énfasis en la calidad del trabajo por encima de la calidad del vestuario. Todos estos son ideales de los hackers, y todos ellos se han infiltrado en el mundo laboral. El tipo de tensión que vi entre hackers y jefes en Sierra On-Line se ha resuelto en gran medida, no sólo en las empresas emergentes, sino también en las más grandes como Google, ya que la mentalidad hacker se ha incorporado como un valor dentro de la empresa. (Por cierto, Ken Williams ha dejado la empresa después de que Sierra fuera absorbida por un conglomerado. "Tanto Roberta como yo hemos 'abandonado' por completo el negocio de los juegos, o incluso el de jugar a juegos", escribe en un correo electrónico. Es un entusiasta de la navegación y ha escrito tres libros sobre sus

aventuras en crucero, y Roberta está trabajando en una novela de no ficción sobre la inmigración irlandesa.)

Ha surgido una nueva generación de hackers, técnicos que no ven a los negocios como un enemigo, sino como el medio a través del cual sus ideas e innovaciones pueden encontrar la audiencia más amplia. Tomemos como ejemplo al director ejecutivo de Facebook, Mark Zuckerberg, que ha logrado que cuatrocientos millones de usuarios compartan sus vidas personales en línea. A sus veinticinco años, ha demostrado ser un maestro en el arte oscuro del desarrollo empresarial, abriendo deliberada y deliberadamente su sitio a anunciantes y vendedores. Sin embargo, él claramente se considera un hacker; el año pasado, dijo a la audiencia en un evento para futuros empresarios de Internet que "Tenemos todo este ethos de que queremos construir una cultura hacker".

Para averiguar qué quería decir con eso, lo visito en la sede de Facebook, un gran edificio en California Avenue en Palo Alto, la misma calle donde alquilé una habitación en 1983 para investigar sobre hackers. Sorprendentemente, el CEO, más conocido por usar polar de North Face, luce corbata. Explica que se acerca el final de un año en el que prometió a su equipo que se presentaría a trabajar con corbata todos los días. Resultó ser un buen año para Facebook: a pesar de la recesión, duplicó su base de usuarios y obtuvo cientos de millones de dólares en ganancias. "Tal vez sea un encanto", dice, sobre la vestimenta. "Pero creo que en su mayoría me ahoga".

El estilo de Zuckerberg puede no provenir de la era dorada del hacking, pero su ética de trabajo sí. "No empezamos con una gran teoría, sino con un proyecto que armamos juntos en un par de semanas", dice Zuckerberg. "Nuestra cultura es que queremos construir algo rápido". Cada seis u ocho semanas, Facebook lleva a cabo "hackatones" donde la gente tiene una noche para soñar y completar un proyecto. "La idea es que se puede construir algo realmente bueno en una noche", dice Zuckerberg. "Y eso es parte de la personalidad de Facebook ahora. Tenemos una gran creencia en avanzar rápido, superar los límites, decir que está bien romper cosas. Definitivamente es algo muy central en mi personalidad".

En la competencia constante por el talento, Zuckerberg cree que la empresa con los mejores hackers gana. "Un buen hacker puede ser tan bueno como diez o veinte ingenieros, y tratamos de adoptar eso. Queremos ser el lugar donde los mejores hackers quieran trabajar, porque nuestra cultura está configurada para que puedan construir cosas rápidamente y hacer cosas locas y puedan ser reconocidos por su brillantez sobresaliente".

A diferencia de los hackers originales, la generación de Zuckerberg no tuvo que empezar desde cero ni usar lenguaje ensamblador para controlar sus máquinas. "Nunca quise desarmar mi computadora", dice. Como hacker en ciernes a fines de los 90, Zuckerberg trasteó con los lenguajes de alto nivel, lo que le permitió concentrarse en los sistemas, en lugar de en las máquinas.

Por ejemplo, cuando jugaba con sus amadas Teenage Mutant Ninja Turtles, Zuckerberg, no representaba guerras con ellas como hacían otros niños. Construía sociedades y pretendía que las Tortugas interactuaban entre sí. "Sólo me interesaba cómo funcionan los sistemas", dice. De manera similar, cuando empezó a jugar con ordenadores, no hackeaba placas madre ni teléfonos, sino comunidades enteras, explotando errores para expulsar a sus amigos de AOL Instant Messenger, por ejemplo.

Como en el caso de Gates, a menudo se acusa a Zuckerberg de dar la espalda a los ideales de los hackers porque se negó a permitir que otros sitios accedieran a la información que aportan los

usuarios de Facebook. Pero Zuckerberg dice que la verdad es exactamente la contraria; su empresa se aprovecha del libre flujo de información y se basa en él. “Nunca tuve esa idea de querer tener información que otras personas no tenían”, dice. “Simplemente pensé que todo debería estar más disponible. El mundo se estaba volviendo más abierto y un mayor acceso a la información era realmente bueno. Por todo lo que leo, eso es una parte muy central de la cultura hacker. Como ‘La información quiere ser libre’ y todo eso”.

Una generación anterior de hackers (y yo) nos preocupó que el mundo del comercio pudiera ahogar la innovación y obstaculizar un movimiento cultural en expansión. Pero el hackerismo ha sobrevivido y prosperado, lo que demuestra su flexibilidad y su poder. Según el editor de libros de informática Tim O’Reilly, que fomenta el hackerismo a través de sus “desconferencias” Foo Camp, la cultura hacker siempre encontrará nuevas salidas. (No es casualidad que esta nueva edición de Hackers esté bajo el sello de O’Reilly). Las grandes empresas pueden tropezar con los avances de los hackers y convertirlos en productos básicos, pero los hackers simplemente se moverán hacia nuevas fronteras. “Es como esa frase de El último tango en París”, dice O’Reilly, “donde Marlon Brando dice: ‘Se acabó, y luego comienza de nuevo’”.

La frontera actual para los hackers, dice O’Reilly, no es el reino puramente matemático de los unos y los ceros, sino las cosas reales: tomar la misma actitud de desarmarlo y construirlo de nuevo que los programadores adoptaron una vez con los compiladores, y aplicarla a las cometas y partes del cuerpo generadoras de energía. (O’Reilly publica la revista Make y organiza los festivales Maker Faire, celebraciones de este espíritu DIY). “El bricolaje es en realidad otra palabra para hackear”, dice. Pero incluso en este ámbito, señala, se ha iniciado el cambio hacia el espíritu emprendedor. O’Reilly dice que ahora la acción está en la biología del bricolaje: manipular el código genético de las células de la misma manera que una generación anterior de hackers manipulaba el código informático. “Todavía está en la etapa divertida”.

Basta con preguntarle a Bill Gates. Si fuera un adolescente de nuevo, estaría hackeando biología. “Creando vida artificial con síntesis de ADN. Eso es una especie de equivalente a la programación en lenguaje de máquina”, dice Gates, cuyo trabajo para la Fundación Bill y Melinda Gates lo ha llevado a convertirse en un experto didáctico en enfermedades e inmunología. “Si quieres cambiar el mundo de alguna manera importante, ahí es donde debes empezar: con las moléculas biológicas. Todos esos son problemas bastante profundos que necesitan el mismo tipo de fanatismo loco de genio juvenil e ingenuidad que impulsó la industria de las PC, y pueden tener el mismo impacto en la condición humana”.

En otras palabras, Gates espera que los hackers también sean los héroes de la próxima revolución. Me parece bien.

—Steven Levy, mayo de 2010

Apéndice D. Notas

La principal fuente de información para Hackers fueron más de cien entrevistas personales realizadas en 1982 y 1983. Además de éstas, hago referencia a varias fuentes escritas.

Primera parte

Capítulo 1 Peter Samson codificó parte de la jerga del TMRC en el inédito "An Abridged Dictionary of the TMRC Language", de alrededor de 1959. Este era aparentemente el núcleo de un diccionario hacker, mantenido en línea en el MIT durante años, que finalmente fue ampliado a The Hacker Dictionary por Gus Steele et al. (Nueva York: Harper & Row, 1983).

Capítulo 1 El poema de Samson impreso en F.O.B., el boletín del TMRC, vol. VI, n.º 1 (septiembre de 1960).

Capítulo 1 “. . . abundaban las historias . . .” Véase Scientific Temperaments: Three Lives in Contemporary Science, de Philip J. Hilts (Nueva York: Simon & Schuster, 1982).

Capítulo 2 Para conocer los antecedentes de IBM, véase The Computer Establishment, de Katharine Davis Fishman (Nueva York: Harper & Row, 1981).

Capítulo 3 Además de las entrevistas personales, se obtuvo cierta información sobre Spacewar del artículo de J.M. Garetz, "The Origin of Spacewar!", en Creative Computing Video and Arcade Games, así como del artículo del mismo autor, "Spacewar: Real-time Capability of the PDP-1", presentado en 1962 ante la Digital Equipment Computer Users' Society, y "Spacewar: Fanatic Life and Symbolic Death Among the Computer Bums", de Stewart Brand, en Rolling Stone, 7 de diciembre de 1972.

Capítulo 3 "Lo que el usuario quiere..." McCarthy citó de su libro Time Sharing Computer Systems (Cambridge, Mass.: MIT Press, 1962).

Capítulo 4 El funcionamiento del juego Peg Solitaire se describe en "Hakmem", de M. Beeler et al. (Massachusetts Institute of Technology, AI Lab Memo No. 239, febrero de 1972).

Capítulo 4 El memorando de Gosper es parte de "Hakmem", mencionado anteriormente.

Capítulo 4 Simon es citado de Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence (San Francisco: W.H. Freeman & Co., 1979), de Pamela McCorduck, un libro que me resultó extremadamente útil para obtener información de fondo sobre los planificadores del laboratorio de IA.

Capítulo 6 El informe de Donald Eastlake fue "ITS Status Report" (Massachusetts Institute of Technology, AI Lab Memo No. 238, abril de 1972).

Capítulo 7 Joseph Weizenbaum, Computer Power and Human Reason (San Francisco: W.H. Freeman & Co., 1976).

Capítulo 7 Bruce Buchanan citado en la “Introducción a la serie de memorandos del Laboratorio de Inteligencia Artificial de Stanford” (Proyecto de Programación Heurística de la Universidad de Stanford, Informe No. HPP-83–25).

Capítulo 7 Además de la columna “Juegos matemáticos” en la revista Scientific American de octubre y noviembre de 1970, Martin Gardner escribe extensamente sobre LIFE de Conway en su libro Wheels, Life, and Other Mathematical Amusements (Nueva York: W.H. Freeman & Co., 1983), que menciona a Gosper de forma destacada.

Segunda parte

Capítulo 8 El mensaje de Benway y otras misivas electrónicas sobre el sistema se encontraron en los extensos álbumes de recortes que Community Memory mantuvo sobre el proyecto. Capítulo 8 Cita de Felsenstein de su “Información biográfica de fondo” de cuatro páginas, fechada el 29 de enero de 1983.

Capítulo 8 Robert A. Heinlein, Revolt in 2100 (Nueva York: Signet, 1954).

Capítulo 8 Un relato en primera persona de las actividades de Albrecht a principios de los años 1960 se encuentra en “A Modern-Day Medicine Show”, Datamation, julio de 1963.

Capítulo 8 “. . . la posibilidad de millones . . .” Véase John Kemeny, Man and the Computer (Nueva York: Scribners, 1972), citado en Robert A. Kahn, “Creative Play with the Computer: A Course for Children”, artículo inédito escrito para el Lawrence Hall of Science. Berkeley, California.

Capítulo 8 “. . . dymaxion . . .” Véase Hugh Kenner, Bucky: A Guided Tour of Buckminster Fuller (Nueva York: Morrow, 1973).

Capítulo 8 Los números anteriores de PCC, generosamente proporcionados por Bob Albrecht, fueron particularmente útiles para obtener información sobre el hacking en el Área de la Bahía a principios de los setenta.

Capítulo 8 Ted Nelson, Computer Lib/Dream Machines (autoeditado, distribuido por The Distributors, South Bend, Ind., 1974).

Capítulo 8 El poema de Brautigan está en The Pill Versus the Springhill Mine Disaster (Nueva York: Dell, Laurel, 1973). Reimpreso con permiso.

Capítulo 8 “. . . un manipulador . . .” William Burroughs en Naked Lunch (Nueva York: Grove Press. 1959).

Capítulo 9 Ivan Illich, Tools for Conviviality (Nueva York: Harper Colophon Books, 1973).

Capítulo 9 “. . .en honor al héroe popular estadounidense . . .” Véase el artículo de Felsenstein. “The Tom Swift Terminal. A Convivial Cybernetic Device”, Journal of Community Communications, junio de 1975.

Capítulo 9 Para obtener información sobre la evolución del microchip y su efecto en Silicon Valley, véase The New Alchemists de Dirk Hansen (Boston: Little, Brown, 1982).

Capítulo 9 “Moore parecía conseguir el dinero . . .” Véase Thomas Albright y Charles Moore, “Las últimas doce horas de toda la Tierra”, Rolling Stone, 8 de julio de 1971. Maureen Orth siguió la historia para Rolling Stone en “Continúa la desaparición de Whole Earth $$$” (16 de marzo de 1972).

Capítulo 9 El folleto se reimprimió en el primer número del Homebrew Computer Club Newsletter (HBCCN), que me resultó invaluable para la investigación en esta sección.

Capítulo 11 El artículo de Pittman se publicó en Actas de la Segunda Feria de Computadoras de la Costa Oeste Jim Warren, ed. (Palo Alto: Computer Faire, 1978).

Capítulo 11 “Cuando [Marsh] tenía poco más…” El artículo de Felsenstein, “Sol: The Inside Story”, apareció en el primer número (julio de 1977) de la efímera revista ROM.

Capítulo 12 El artículo de Esquire. “Secrets of the Black Box”, de Ron Rosenbaum, se reimprime en su Rebirth of the Salesman: Tales of the Song and Dance 70’s (Nueva York: Delta, 1979).

Capítulo 12 “El ganar no es tan importante…” Una entrevista inédita con el periodista Doug Garr.

Capítulo 12 Parte de la información sobre Draper fue extraída de Fighting Computer Crime de Donn Parker (Nueva York: Scribners, 1983).

Capítulo 12 “La barba de Fidel Castro…” .” Véase Paul Ciotti, "Revenge of the Nerds", California, julio de 1982.

Capítulo 12 “Prepárense para el despegue...” Véase Elizabeth Fairchild, “The First West Coast Computer Faire”, ROM, julio de 1977.

Capítulo 12 El discurso de Nelson se reimprime en The First West Coast Computer Faire Proceedings, Jim Warren, ed. (Palo Alto: Computer Faire, 1977).

Tercera parte

Capítulo 14 Los Carpetbaggers (Nueva York: Pocket Books, 1961).

Capítulo 14 La carta se publicó en Purser’s Magazine, invierno de 1981.

Capítulo 15 “Un participante explicó más tarde a un periodista...” El reportero era de Softline, otra publicación de Tommervik, que empezó con fondos de los Williams. Tanto Softline como Softalk proporcionaron considerable información de fondo sobre la Hermandad.

Capítulo 16 “. . . diseñadores de toallas . . .” Véase John F. Hubner y William F. Kistner, “What Went Wrong at Atari?”—un artículo reimpreso en InfoWorld, 28 de noviembre de 1983 y 5 de diciembre de 1983. Otros antecedentes sobre Atari de Video Invaders de Steve Bloom (Nueva York: Arco, 1982).

Capítulo 19 “. . . entrevistado en un artículo . . .” Véase Lee Gnomes, “Secrets of the Software Pirates”, Esquire, enero de 1982.

Apéndice A “no significa . . .” Stallman almacenó varias “llamas” (escritos apasionados) en el sistema informático del MIT, entre ellas “Ensayo”, “Gnuz” y “Wiezenbomb”. La cita es de su “Ensayo” autobiográfico.

Apéndice A “que me devuelvan todas las extensiones…” Del “Ensayo” de Stallman.

Apéndice A “Es doloroso para mí…” “Ensayo”.

Apéndice E. Agradecimientos

Estoy en deuda con muchas personas que me ayudaron de diversas maneras mientras trabajaba en Hackers. Primero, con las personas que aceptaron ser entrevistadas para el libro. Algunos eran veteranos de este tipo de intercambio periodístico; otros solo habían hablado con entrevistadores sobre cuestiones técnicas y nunca antes habían hablado de la naturaleza personal o filosófica del hacking; otros simplemente no habían hablado con personas como yo. Casi todos hablaron con libertad y franqueza; creo que no es coincidencia que los hackers sean tan libres en la conversación, una vez que comienzan, como lo son al compartir código informático. Muchos de los siguientes aceptaron múltiples entrevistas y, a menudo, llamadas de seguimiento para verificar hechos o aclarar detalles técnicos.

Mis conversaciones con ellos fueron la columna vertebral del libro, y me gustaría agradecer, en orden alfabético, a Arthur Abraham, Roe Adams, Bob Albrecht, Dennis Allison, Larry Bain, Alan Baum, Mike Beeler, Dorothy Bender, Bill Bennett, Chuck Benton, Bob y Carolyn Box, Keith Britton, Lois Britton, Bill Budge, Chuck Bueche, David Bunnell, Doug Carlston, Gary Carlston, Marie Cavin, Mary Ann Cleary, Bob Clements, Tracy Coats, David Crane, Edward Currie, Rick Davidson, Bob Davis, Jack Dennis, Peter Deutsch, Steve Dompier, John Draper, Dan Drew, Mark Duchaineau, Les Earnest, Don Eastlake, Doug Englebart, Chris Espinosa, Lee Felsenstein, LeRoy Finkel, Howard Franklin, Bob Frankston, Ed Fredkin, Gordon French, Martin Garetz, Harry Garland, Richard Garriott, Lou Gary, Bill Gates, Bill Godbout, Vincent Golden, Dave Gordon, Ralph Gorin, Dan Gorlin, Bill Gosper, Richard Greenblatt, Margaret Hamilton, Eric Hammond, John Harris, Brian Harvey, Ted Hoff, Kevin Hunt, Chris Iden, Jerry Jewell, Robert Kahn, David Kidwell, Gary Kildall, Tom Knight, Joanne Koltnow, Alan Kotok, Marc LeBrun, Bob Leff, Mike Levitt, Efrem Lipkin, David Lubar, Olaf Lubeck, John McCarthy, John McKenzie, Robert Maas, Patricia Mariott, Bob Marsh, Roger Melen, Jude Milhon, Marvin Minsky, Fred Moore, Stewart Nelson, Ted Nelson, Jim Nitchals, Russ Noftsker, Kenneth Nussbacher, Rob O'Neal, Peter Olyphant, Adam Osborne, Bill Pearson, Tom Pittman, Larry Press, Malcolm Rayfield, Robert Reiling, Randy Rissman, Ed Roberts, Steve Russell, Peter Samson, Bob Saunders, Warren Schwader, Gil Segal, Vic Sepulveda, David Silver, Dan Sokol, Les Solomon, Marty Spergel, Richard Stallman, Jeff Stephenson, Ivan Strand, Jay Sullivan, Dick Sunderland, Gerry Sussman, Tom Tatum, Dick Taylor, Robert Taylor, Dan Thompson, Al Tommervik, Margot Tommervik, Mark Turmell, Robert Wagner, Jim Warren, Howard Warshaw, Joseph Weizenbaum, Randy Wigginton, John Williams, Ken Williams, Roberta Williams, Terry Winograd, Donald Woods, Steve Wozniak y Fred Wright.

Quisiera agradecer especialmente a las personas mencionadas anteriormente que me brindaron una atención extraordinaria, entre las que se incluyen (aunque no se limitan a) Lee Felsenstein, Bill Gosper, Richard Greenblatt, Peter Samson, Ken Williams y Roberta Williams. Durante el curso de mi investigación, me beneficié de la hospitalidad de instituciones como la Biblioteca de Ciencias de la Computación del MIT, la Biblioteca de Stanford, el Museo de la Computación, el Salón de Ciencias Lawrence y la Biblioteca de la Universidad de California.

En mis viajes a California y Cambridge, me beneficié de la hospitalidad de Phyllis Coven, Art Kleiner, Bill Mandel y John Williams. Lori Carney y otros mecanografiaron miles de páginas de transcripciones. La exigente edición de Viera Morse me permitió mantener la honestidad lingüística. Los editores de revistas David Rosenthal y Rich Friedman me dieron trabajo que me permitió seguir

adelante. Mis colegas redactores informáticos Doug Garr, John Markoff, Deborah Wise y los miembros del Lunch Group me dieron buenos consejos. Mis padres, mi hermana Diane Levy, mis amigos Larry Barth, Bruce Buschel, Ed Kaplan, William Mooney, Randall Rothenberg, David Weinberg y muchos otros (ellos saben quiénes son) que tendrán que aceptar esta mención insuficiente.

El libro también fue producto del entusiasmo y la paciencia de mi agente, Pat Berens, y de mi editor, James Raimes, quienes me alentaron enormemente. Esos términos también se aplican a Teresa Carpenter, que se las arregló magníficamente con el libro y su autora durante el largo proceso de investigación y redacción.

Por último, gracias a Steve Wozniak por diseñar esa Apple II en la que escribí el libro. Si no hubiera sido por la revolución que abordo en Hackers, mis trabajos podrían haber continuado durante otro año, sólo para obtener un borrador limpio de mi máquina de escribir.

Apéndice F. Acerca del autor

Steven Levy es un escritor senior de Wired. Anteriormente, fue escritor jefe de tecnología y editor senior de Newsweek. Levy ha escrito seis libros y ha publicado artículos en Harper's, Macworld, New York Times Magazine, New Yorker, Premiere y Rolling Stone. Steven ha ganado varios premios durante sus más de 30 años de escribir sobre tecnología, incluyendo **Hackers**, que PC Magazine nombró el mejor libro de ciencia y tecnología escrito en los últimos veinte años, y Crypto, que ganó el gran premio de libro electrónico en el Festival del Libro de Frankfurt de 2001.

Índice

Una nota sobre el índice digital

Un enlace en una entrada de índice se muestra como el título de la sección en la que aparece esa entrada. Debido a que algunas secciones tienen múltiples marcadores de índice, no es inusual que una entrada tenga varios enlaces a la misma sección. Al hacer clic en cualquier enlace, accederá directamente al lugar del texto en el que aparece el marcador.

Símbolos

25th Hacker's Conference, Afterword: 2010
8080 chip, Tiny BASIC

A

Acculators, Greenblatt and Gosper
Adams, Roe, Summer Camp
ADD (command), Revolt in 2100
Advanced Research Project Agency (ARPA), Spacewar, Life, Life, Life
Adventure Development Language (ADL), The Wizard and the Princess, Summer Camp
Adventure game, Life
Adventure-style games, The Wizard and the Princess, The Wizard and the Princess, The Brotherhood
Albrecht, Bob, Revolt in 2100, Revolt in 2100, Every Man a God, The Homebrew Computer Club, Tiny BASIC
Alchemy and Artificial Intelligence, Greenblatt and Gosper
Algorithm, The Hacker Ethic
Alkabeth game, Applefest
Allen, Paul, Tiny BASIC, Tiny BASIC, Tiny BASIC
Allison, Dennis, Tiny BASIC
Altair, Every Man a God, Every Man a God, Every Man a God, The Homebrew Computer Club, The Homebrew Computer Club, The Homebrew Computer Club
Altair BASIC, Tiny BASIC
Altair Users Newsletter, Tiny BASIC
Antic chip (Atari), The Third Generation
Apollo 17 moon shot, Life
Apple computer, The Wizard and the Princess, The Wizard and the Princess, The Brotherhood, The Brotherhood, The Third Generation, Summer Camp, Frogger, Frogger, Applefest, Wizard vs. Wizards, Wizard vs. Wizards
Apple Computer Company, Woz, Woz, Secrets, The Brotherhood, Afterword: 2010
Apple II computer, Woz, Woz, The Wizard and the Princess, Summer Camp, Afterword: 2010
    design, Woz
Apple World, The Brotherhood, The Brotherhood
Applefest, Applefest, Applefest
Arcade games, The Brotherhood
Art of Computer Programming, The (Knuth), Life
Artificial intelligence (AI), The Tech Model Railroad Club, The Hacker Ethic, Life, Life
    AI lab, Life, Life
Asimov, Isaac, The Midnight Computer Wiring Society, Life, Life
Assembly computer language, The Third Generation
Assembly language, The Tech Model Railroad Club

Assembly-language programmers, The Third Generation
Atari (game company), The Homebrew Computer Club, Woz, The Brotherhood, The Third Generation, The Third Generation, The Third Generation, The Third Generation, The Third Generation, The Third Generation, The Third Generation, The Third Generation, Frogger, Frogger, Wizard vs. Wizards, Wizard vs. Wizards, Wizard vs. Wizards
    Atari 400, The Third Generation, The Third Generation
    Atari 800, The Third Generation, The Third Generation, The Third Generation, The Third Generation, Wizard vs. Wizards
    Pac-Man injunction, The Third Generation
    VCS game machine, The Third Generation
    VCS machine, Frogger, Wizard vs. Wizards
Awake! publication, Summer Camp

B

Baba Ram Das, Revolt in 2100
Bachelor mode, The Midnight Computer Wiring Society
Bain, Larry, Summer Camp
BAL computer language, The Wizard and the Princess
BASIC computer language, Spacewar, Revolt in 2100, Revolt in 2100, Tiny BASIC, Tiny BASIC, Tiny BASIC, Woz
    Tiny BASIC, Tiny BASIC, Tiny BASIC
Battle Warp game, The Third Generation
Baum, Alan, The Homebrew Computer Club, Woz, Woz, Woz
BC cartoon, Wizard vs. Wizards
Beeler, Mike, Greenblatt and Gosper, Life
Beer Run game, Frogger
Bekins Moving and Storage, The Wizard and the Princess
Bell System Technical Journal, The Midnight Computer Wiring Society
Bell Telephone System, The Midnight Computer Wiring Society
Bell, Gordon, Greenblatt and Gosper
Bennett, Bill, The Midnight Computer Wiring Society
Benway, Doctor, Revolt in 2100, Revolt in 2100
Berkeley Barb, Revolt in 2100
Berzerk, Frogger
Bill and Melinda Gates Foundation, Afterword: 2010
Binary number system, The Tech Model Railroad Club
Bit (measurement), The Tech Model Railroad Club
Bit Works, Applefest
Blattisms, Greenblatt and Gosper
Blue box long-distance calls, Secrets
Blue boxes, The Midnight Computer Wiring Society, Woz
Bolt Beranek and Newman (BBN), Spacewar, Spacewar
Bootstrap, The Homebrew Computer Club
Box, Bob and Carolyn, Frogger, Wizard vs. Wizards
Brand, Stewart, Every Man a God, Afterword: 2010
Brautigan, Richard, Revolt in 2100
Broken Flowers (movie), Afterword: 2010
Brotherhood (Brøderbund) Software, The Brotherhood, The Brotherhood, The Brotherhood, Frogger, Wizard vs. Wizards
Buchanan, Bruce, Life
Budge, Bill, Summer Camp

Bueche, Chuck, Frogger, Applefest, Applefest
Bug (robot) program, Winners and Losers, Winners and Losers
Bunnell, David, Every Man a God
Burnstein, Malcolm, Revolt in 2100
Burroughs computer, The Wizard and the Princess
Byte (measurement), Every Man a God
Byte magazine, Tiny BASIC, Tiny BASIC
Byte Shop, Applefest

C

California Polytechnic, Pomona Campus, The Wizard and the Princess
Call Computer service, Woz
Cambridge urchins, The Tech Model Railroad Club
Captain Crunch, Woz, Woz, Secrets, The Brotherhood, Applefest
Carlston, Doug, The Brotherhood, The Brotherhood, Applefest
Carlston, Gary, The Brotherhood, The Brotherhood
Carmack, John, Afterword: 2010
Carnegie-Mellon University, Life, Life
Carpetbaggers, The (Robbins), The Wizard and the Princess
Cassette recorders, The Brotherhood
CBS opening, Spacewar
Charles Adams Associates, Greenblatt and Gosper, Greenblatt and Gosper
Chat program, Woz
Chess program, The Tech Model Railroad Club, The Hacker Ethic
Chicken Hawk system, Every Man a God
Chinese food, Greenblatt and Gosper
Choplifter game, Frogger
Christie, Agatha, The Wizard and the Princess
Circle Algorithm hack, Spacewar
CL 9, Afterword: 2010
Clements, Bob, The Midnight Computer Wiring Society
Clue board game, The Wizard and the Princess
Coarsegold, The Wizard and the Princess, The Third Generation
Coats, Tracy, Wizard vs. Wizards
Commodore computer, Secrets, The Brotherhood, The Third Generation
Communist Party, Revolt in 2100, Revolt in 2100
Community Memory, Revolt in 2100, Revolt in 2100, Revolt in 2100, Every Man a God, Every Man a God, Tiny BASIC, Woz, Secrets, Secrets, Secrets
Compatible Time-sharing System (CTSS), Winners and Losers
Compiler, The Hacker Ethic
Computer Faire, Woz, Secrets, Secrets, The Brotherhood, The Third Generation
Computer Lib (Nelson), Revolt in 2100, Every Man a God, Every Man a God, Woz
Computer Power and Human Reason (Weizenbaum), Life
Computerland, Secrets
Computers, effect on people’s lives, The Hacker Ethic, Summer Camp
Conan the Barbarian (movie), Applefest
Cons (conferences), The Third Generation
Consumer Electronics Shows, Applefest
Continental Company, The Brotherhood
Control Data Company, Revolt in 2100
Control Data Institute, The Wizard and the Princess

Conway, John, Life, The Homebrew Computer Club
Copy protection, Applefest
Corbató, F. J., Spacewar, Winners and Losers
Crawford, Chris, The Third Generation
Crazy Climber game, The Third Generation
Creepy Corridors game, Frogger
Cribbage computer game, Summer Camp
Cromemco, The Homebrew Computer Club, Tiny BASIC
Crowther, Will, Life
Cruft, The Tech Model Railroad Club
Currie, Eddie, Every Man a God

D

DAC (Deposit Accumulator), Spacewar
Daisy (song), The Homebrew Computer Club
Dancing with the Stars (TV show), Afterword: 2010
Dark Crystal (movie), Frogger
Data General Nova computer, Woz
DataSoft, Frogger
Davidson, Rick, Summer Camp
Davidson, Sharon, Summer Camp
Davis, Bob, Summer Camp, Wizard vs. Wizards
Dazzler, The Homebrew Computer Club, Tiny BASIC
DECAL assembler, The Midnight Computer Wiring Society
Decimal print program, The Hacker Ethic
Dennis, Jack, The Tech Model Railroad Club, Spacewar, Spacewar
Department of Defense, Life
Deus Ex Machina, or The True Computerist, Tiny BASIC
Deutsch, Peter, The Tech Model Railroad Club, Spacewar, Spacewar, Revolt in 2100
Dial-a-joke service, Woz
Dictionary program, Frogger
Digital Equipment Corporation (DEC), Spacewar, Spacewar, Winners and Losers, Revolt in 2100
Digital rights management (DRM), Afterword: 2010
Disassemblers, The Third Generation
Disk drives, The Brotherhood
Display hacks, Spacewar
Display list interrupts, The Third Generation
Dithering (color technique), The Wizard and the Princess
DIY biology, Afterword: 2010
DNA synthesis, Afterword: 2010
Docusports, Wizard vs. Wizards
Dompier, Steve, Every Man a God, The Homebrew Computer Club, The Homebrew Computer Club, The Homebrew Computer Club, The Homebrew Computer Club, Tiny BASIC, Tiny BASIC, Tiny BASIC
Doom game, Afterword: 2010
Dr. Dobbs Journal (DDJ), Tiny BASIC, Woz, Woz
Draper, John, Woz, Secrets, The Brotherhood, Applefest
Dreyfus, Herbert, Greenblatt and Gosper
Duchaineau, Mark, Applefest, Applefest
Dungeons and Dragons, Summer Camp, Applefest
Dymax, Revolt in 2100

E

E-Z-Draw program, The Brotherhood
EAM room, The Tech Model Railroad Club, The Tech Model Railroad Club
Earth Dies Screaming game, Frogger
Eastlake, Don, Winners and Losers
Easy Writer program, The Brotherhood, Applefest
Edwards, Dan, Spacewar, The Midnight Computer Wiring Society
Electrical engineering course No. 641, The Tech Model Railroad Club
Electronic Accounting Machinery (EAM), The Tech Model Railroad Club
Electronic Arts, Applefest
ELlZA program, Life, Every Man a God
EMACS editing program, The Last of the True Hackers
Erhard, Werner, Tiny BASIC
Espinosa, Chris, Woz, Woz, Secrets
Esquire magazine, Woz, Applefest
Est training, Tiny BASIC
Expensive Desk Calculator program, The Hacker Ethic
Expensive Typewriter program, The Hacker Ethic, Frogger

F

F&T Diner, Greenblatt and Gosper
Facebook, Afterword: 2010
Felsenstein, Joe, Revolt in 2100, Revolt in 2100
Felsenstein, Lee, Revolt in 2100, Every Man a God, Every Man a God, Every Man a God, The Homebrew Computer Club, The Homebrew Computer Club, The Homebrew Computer Club, The Homebrew Computer Club, The Homebrew Computer Club, Tiny BASIC, Woz, Secrets, Secrets, Secrets, Afterword: 2010
FIND (command), Revolt in 2100
Finkel, LeRoy, Revolt in 2100
Flexowriter, The Tech Model Railroad Club, The Tech Model Railroad Club
FLIT, The Tech Model Railroad Club, Spacewar
    Micro-FLIT, Spacewar
Foo, Winners and Losers
Foo Camp, Afterword: 2010
Fool on the Hill (Beatles ballad), The Homebrew Computer Club
Forbin Project (movie), Every Man a God
Form Master (disk-copying machine), Applefest
FORTRAN computer language, The Hacker Ethic, Greenblatt and Gosper, Revolt in 2100, The Wizard and the Princess, The Wizard and the Princess
Fountainhead, The (Rand), Winners and Losers
Franklin, Benjamin, Afterword: 2010
Fredkin, Ed, The Midnight Computer Wiring Society, Winners and Losers, Life, Life, Life, Life
Free Software Foundation, Afterword: 2010
Free Speech Movement, Revolt in 2100
French, Gordon, Every Man a God, The Homebrew Computer Club, The Homebrew Computer Club
Fresno County jail, Wizard vs. Wizards
Fried, Ben, Afterword: 2010
Friedman, Barry, Wizard vs. Wizards
Frogger game, Frogger, Applefest, Wizard vs. Wizards, Wizard vs. Wizards
Full-screen editing, The Third Generation

Fuller, Buckminster, Revolt in 2100, Revolt in 2100, Every Man a God, The Homebrew Computer Club
Fusion-io, Afterword: 2010

G

Galactic Saga game, The Brotherhood
Galaxian game, The Brotherhood, Summer Camp
Gamma Scientific company, The Third Generation, The Third Generation
Gardner, Martin, Life
Garetz, Shag, Spacewar
Garland, Harry, The Homebrew Computer Club, The Homebrew Computer Club
Garriott, Owen K., Applefest
Garriott, Richard, Applefest, Applefest, Applefest
Gates, Bill, Tiny BASIC, Tiny BASIC, Frogger, Afterword: 2010, Afterword: 2010, Afterword: 2010
Gebelli, Nasir, The Brotherhood, Frogger
General Electric Science Fair, The Tech Model Railroad Club
George Jackson People's Free Clinic, Revolt in 2100
Glider gun, Life
GNU operating system, Afterword: 2010
Gobbler game, The Third Generation
Godbout, Bill, Every Man a God
Going public, The Wizard and the Princess
Gold panning, Frogger
Golden, Vinnie "the Bear", Every Man a God
Google, Afterword: 2010
Gorlin, Dan, Frogger
Gosper, Bill, Spacewar, Greenblatt and Gosper, Greenblatt and Gosper, Greenblatt and Gosper, Winners and Losers, Winners and Losers, Life, Life, Afterword: 2010
Graham, Paul, Afterword: 2010
Great Subway Hack, Winners and Losers, Life
Greenblatt, Richard, Greenblatt and Gosper, Greenblatt and Gosper, Greenblatt and Gosper, Greenblatt and Gosper, Winners and Losers, Winners and Losers, Life, Life, Life, Life, The Wizard and the Princess, Afterword: 2010
Griffin, Kathy, Afterword: 2010
Gronk, Winners and Losers
Guinness Book of World Records, Frogger
Gulf & Western Conglomerate, Sega division, Frogger

H

Hack, The Tech Model Railroad Club
Hackathons, Afterword: 2010
HACKER (Sussman program), Winners and Losers
Hacker Dojo, Afterword: 2010
Hacker Ethic, The Hacker Ethic, The Hacker Ethic, Spacewar, Greenblatt and Gosper, The Midnight Computer Wiring Society, Winners and Losers, Life, Life, Revolt in 2100, Revolt in 2100, Every Man a God, Every Man a God, The Homebrew Computer Club, The Homebrew Computer Club, Tiny BASIC, Secrets, Secrets, The Wizard and the Princess, The Brotherhood, The Brotherhood, The Brotherhood, The Third Generation, Frogger, Frogger, Applefest, Applefest, Wizard vs. Wizards

Hackers, The Tech Model Railroad Club, Greenblatt and Gosper, Life, Every Man a God, Wizard vs. Wizards
hardware, Life
Hackers Conference, Afterword: 2010
Hacking, current state, Afterword: 2010
Haight-Ashbury Medical Research Clinic, Revolt in 2100
HAL (2001), The Homebrew Computer Club
Hamilton, Margaret, The Midnight Computer Wiring Society
Hands-On Imperative, The Tech Model Railroad Club, The Hacker Ethic, Greenblatt and Gosper, Revolt in 2100, Revolt in 2100, The Third Generation
Hardware hackers, Life
Harris, John, The Third Generation, The Third Generation, The Third Generation, The Third Generation, Summer Camp, Summer Camp, Frogger, Frogger, Frogger, Wizard vs. Wizards, Wizard vs. Wizards, Wizard vs. Wizards
Harvard University, Spacewar
Harvey, Brian, Winners and Losers, Life
Hawkins, Trip, Applefest
Haystack Observatory, The Midnight Computer Wiring Society
Hazeltine 1500 terminal, Revolt in 2100
Heath Company, Revolt in 2100
Heffron, Eli, The Tech Model Railroad Club, Winners and Losers
Heinlein, Robert, The Midnight Computer Wiring Society, Revolt in 2100, Every Man a God
Henson, Jim, Frogger
Hertzfeld, Andy, Afterword: 2010
Herzfeld, Andy, The Brotherhood
Heuer, Roberta, The Wizard and the Princess (see)
Hewlett-Packard (HP), Revolt in 2100, Woz, Woz
Hexagon House, The Third Generation, Applefest
Hi-Res Adventure, The Wizard and the Princess
High-level computer language, The Hacker Ethic
Higham Institute, Spacewar, Spacewar
Holm, Celeste, Revolt in 2100
Holt, Rod, Woz
Homebrew Computer Club, Every Man a God, The Homebrew Computer Club, Tiny BASIC, Tiny BASIC, Tiny BASIC, Tiny BASIC, Woz, Woz, Woz, Secrets, Secrets, Afterword: 2010
Honeywell company, Winners and Losers
Hulking Giant, The Tech Model Railroad Club, The Tech Model Railroad Club, Greenblatt and Gosper
Hyperspace, Spacewar

I

IBM (International Business Machines), The Tech Model Railroad Club, The Tech Model Railroad Club, The Tech Model Railroad Club, The Hacker Ethic, The Hacker Ethic, Greenblatt and Gosper, Frogger, Frogger, Wizard vs. Wizards, Wizard vs. Wizards
IBM 704, The Tech Model Railroad Club, The Tech Model Railroad Club, The Hacker Ethic
IBM 709, The Tech Model Railroad Club
PC machines, Frogger
IC (integrated circuit), Every Man a God
Illich, Ivan, Every Man a God, Every Man a God
Image subtraction, Winners and Losers
IMSAI, Tiny BASIC, Woz, Secrets

Incompatible Time-sharing System (ITS), Winners and Losers, Life, Life, Life
Infocom, Summer Camp
Informatics, The Wizard and the Princess, Frogger
Information International Incorporated (Triple-I), The Midnight Computer Wiring Society, Life
Information Should Be Free, Afterword: 2010
Ingram, Gary, Every Man a God, The Homebrew Computer Club, Woz, Secrets
Integer BASIC, Woz
Intel, Every Man a God, Every Man a God, Every Man a God, Every Man a God, Tiny BASIC, Tiny BASIC
    4004 chip, Tiny BASIC, Tiny BASIC
    8008 chip, Every Man a God, Every Man a God
    8080 chip, Every Man a God
Intellectual property law, Afterword: 2010
Intentional system crashes, Winners and Losers
Interface Age magazine, Tiny BASIC
Interval Research, Afterword: 2010
Interzone, Revolt in 2100
Introduction to Computer Progamming (EE 641), Greenblatt and Gosper
iPad, Afterword: 2010
iPhone, Afterword: 2010, Afterword: 2010
iPod, Afterword: 2010
Iran hostage crisis, Frogger

J

Jawbreaker 2, Wizard vs. Wizards
Jawbreaker game, The Third Generation, Frogger
Jehovah's Witnesses, Summer Camp
Jensen, Mr., The Hacker Ethic
Jewell, Jerry, The Brotherhood, The Brotherhood, Frogger, Applefest, Wizard vs. Wizards, Wizard vs. Wizards
Jobs, Steve, Woz, Woz, Woz, Woz, Woz, Secrets
Joplin, Janis, Revolt in 2100
Journal of Community Communications, The Homebrew Computer Club
Joy Fong restaurant, Greenblatt and Gosper
JUMP (command), Spacewar

K

K1uge room, Spacewar
Kahn, Bob, Revolt in 2100, Every Man a God, Woz
Kaleidoscope program, The Homebrew Computer Club
Kemeny, John, Revolt in 2100
Keypunch machine 407, The Tech Model Railroad Club
KILL SYSTEM command, Winners and Losers
Kindle, Afterword: 2010
Kluge room, The Hacker Ethic, Greenblatt and Gosper, Greenblatt and Gosper
Knight, Tom, Winners and Losers, Winners and Losers, Winners and Losers, Life, Life
Knuth, Dona1d, Life
Koltnow, Joanne, Woz
Kotok, Alan, The Tech Model Railroad Club, The Hacker Ethic, Spacewar, Spacewar, Spacewar, Greenblatt and Gosper

L

Lancaster, Don, Every Man a God, Every Man a God, Tiny BASIC
Lawrence Hall of Science, Every Man a God
Leary, Timothy, Revolt in 2100
LeBrun, Marc, Woz
Leopold's Records, Revolt in 2100, Revolt in 2100, Secrets, Secrets
Lessig, Lawrence, Afterword: 2010
Lester (engineering student, University of Missouri), Greenblatt and Gosper
Letter to Hobbyists, Afterword: 2010
Levitt, Mike, Life
LGC Engineering Company, Every Man a God
Lincoln Lab, The Tech Model Railroad Club, The Tech Model Railroad Club, Spacewar, The Midnight Computer Wiring Society
Linux, Afterword: 2010
LIO (Load Input-Output), Spacewar
Lipkin, Efrem, Revolt in 2100, Revolt in 2100, Revolt in 2100, Secrets, Secrets
LISP programming, The Tech Model Railroad Club, Spacewar, Greenblatt and Gosper
Lock hacking, The Midnight Computer Wiring Society
Locksmith program, Applefest
Lode Runner game, Wizard vs. Wizards
LOGO computer language, Life
Long-distance blue box calls, Woz, Secrets, Secrets
Long-Range Computer Study Group, Spacewar
Lord British, Applefest, Applefest
Losers and winners, Winners and Losers
Losing, The Tech Model Railroad Club
Loving Grace Cybernetics, Revolt in 2100
Lubeek, Olaf, The Third Generation

M

M&R Electronics, Woz
MacHack chess program, Greenblatt and Gosper
Machine Aided Cognition, Spacewar
Machine language, The Hacker Ethic
MacLISP, Greenblatt and Gosper
Mailer, Norman, Life
Make magazine, Afterword: 2010
Maker Faire festivals, Afterword: 2010
Mariott, Pat, Applefest
Mark 4, The Wizard and the Princess
Marketing, Frogger
Markkula, Mike, Woz, Secrets
Marsh, Bob, Every Man a God, Every Man a God, The Homebrew Computer Club, The Homebrew Computer Club, Tiny BASIC, Tiny BASIC, Tiny BASIC, Woz, Secrets
Marx, Karl, Every Man a God
Massachusetts Institute of Technology (MIT), The Tech Model Railroad Club, The Hacker Ethic, The Midnight Computer Wiring Society, Life
Master key, The Midnight Computer Wiring Society
Mayan calendar hack, Spacewar
McCarthy, John, The Tech Model Railroad Club, The Hacker Ethic, Spacewar, Spacewar, Greenblatt and Gosper, Life, Life, Revolt in 2100

McKenzie, John, The Tech Model Railroad Club, The Midnight Computer Wiring Society
Melen, Roger, The Homebrew Computer Club, The Homebrew Computer Club
Memorial Student Union (UM), Greenblatt and Gosper
Memory boards, The Homebrew Computer Club, The Homebrew Computer Club
    design and production, The Homebrew Computer Club
Merton, Louis, Life
MICRO magazine, The Wizard and the Princess
Micro-8 Newsletter, Tiny BASIC
Microbooth, The Brotherhood
Microprocessor, Every Man a God
Microsoft, Frogger, Afterword: 2010
MIDAS assembler, The Midnight Computer Wiring Society, Winners and Losers
Middle Earth trilogy, Life
Midnight Computer Wiring Society, The Midnight Computer Wiring Society
Midpeninsula Free University, Revolt in 2100
Milhon, Jude, Revolt in 2100, Revolt in 2100, The Homebrew Computer Club, Secrets, Secrets
Minsky, Marvin, The Hacker Ethic, Spacewar, Spacewar, Greenblatt and Gosper, The Midnight Computer Wiring Society, Winners and Losers, Life, Life
Minskytron, Spacewar, Spacewar
Missile Command game, The Third Generation
MIT hackers, Afterword: 2010
MITS Caravan, Tiny BASIC
Moby Memory, Winners and Losers
Model 33 teletype, Winners and Losers
Model Instrumentation Telemetry System (MITS), Every Man a God, Every Man a God, Every Man a God, Tiny BASIC, Tiny BASIC, Tiny BASIC, Woz
Modem, Secrets
Modes, Greenblatt and Gosper
Monopoly game, Frogger
Moore, Fred, Every Man a God, The Homebrew Computer Club, The Homebrew Computer Club, The Homebrew Computer Club, Secrets
Morby, Jacky, Frogger
Motorola 6800 chip, Tiny BASIC
Motorola microprocessors, Woz
Mouse in the Maze game, Spacewar, The Wizard and the Princess
Mouskattack game, The Third Generation
Multi-format disk scheme, Applefest
Multics, Winners and Losers, Winners and Losers
Multiple Access Computing, Spacewar
Mung (Mash Until No Good), The Tech Model Railroad Club
Muppets, The, Frogger
Music program, The Tech Model Railroad Club, The Hacker Ethic, Spacewar
Music, of a Sort, The Homebrew Computer Club
My Computer Likes Me (Albrecht), Revolt in 2100
Mystery House game, The Wizard and the Princess, The Brotherhood, Wizard vs. Wizards

N

Naked Lunch (Burroughs), Revolt in 2100
NASA, Life
National Strike Information Center, Life
NBC Magazine, Frogger

Nelson, Ralph, Revolt in 2100
Nelson, Stewart, The Midnight Computer Wiring Society, The Midnight Computer Wiring Society, The Midnight Computer Wiring Society, The Midnight Computer Wiring Society, Life, Every Man a God
Nelson, Ted, Revolt in 2100, Every Man a God, The Homebrew Computer Club, Woz, Secrets, Frogger
Newton-John, Olivia, Summer Camp
Noftsker, Russell, Greenblatt and Gosper, The Midnight Computer Wiring Society, Life
Nomography, Greenblatt and Gosper
Notes on Distant Dialing, The Midnight Computer Wiring Society
Nussbacher, Ken, The Third Generation

O

Oakhurst, The Wizard and the Princess, The Wizard and the Princess, Summer Camp
Oakland Seven, Revolt in 2100
Oasis, The Homebrew Computer Club, Tiny BASIC
Octal number system, The Tech Model Railroad Club
Officially Sanctioned Programs, Winners and Losers
Officially Sanctioned Users, The Tech Model Railroad Club, Winners and Losers
On-Line Systems, The Wizard and the Princess, The Wizard and the Princess, The Brotherhood, The Brotherhood, The Brotherhood, The Third Generation, The Third Generation, Summer Camp, Summer Camp, Frogger, Frogger, Frogger, Frogger, Frogger, Wizard vs. Wizards
    (see also)
    programming office, The Wizard and the Princess
    recruiting, Summer Camp
Open Letter to Hobbyists, Tiny BASIC
Orifice, The Tech Model Railroad Club
Osborne 1 computer, Secrets
Osborne, Adam, Secrets, Secrets
O’Reilly, Tim, Afterword: 2010

P

Pac-Man, The Third Generation, The Third Generation, Frogger
Page flipping, The Brotherhood
Papert, Seymour, Life
Parker Brothers, Frogger, Wizard vs. Wizards
Password hacking, Winners and Losers
Password TV gameshow, The Brotherhood
PC ’76 computer show, Tiny BASIC
PDP-1, Spacewar, Spacewar, Greenblatt and Gosper, Greenblatt and Gosper, The Midnight Computer Wiring Society, The Midnight Computer Wiring Society, The Wizard and the Princess
PDP-10, Winners and Losers, Life, Life
PDP-6, Greenblatt and Gosper, Greenblatt and Gosper, Life
    PDP-6 MacLISP, Greenblatt and Gosper
PDP-7, The Midnight Computer Wiring Society
PDP-8, Revolt in 2100
Peanut (PCjr) home computer, Frogger
Peg Solitaire (HI-Q), Greenblatt and Gosper
Pennywhistle modem, The Homebrew Computer Club, Secrets
People magazine, Wizard vs. Wizards

People's Computer Company, Revolt in 2100, Revolt in 2100, Every Man a God, Every Man a God, Every Man a God, The Homebrew Computer Club, Tiny BASIC
People's Park, Every Man a God
Personal computers, effect on people's lives, The Hacker Ethic, Summer Camp
PET computer, The Third Generation, The Third Generation
Phone company, Spacewar, The Midnight Computer Wiring Society, Woz
phone hacking, The Midnight Computer Wiring Society
phone phreaks, Woz
Ping-Pong, Winners and Losers, Life
Pittman, Tom, The Homebrew Computer Club, Tiny BASIC, Woz
PL/I computer language, Winners and Losers
Plank House in the Valley, The Wizard and the Princess
Playability (computer games), The Third Generation
Player-missile graphics, The Third Generation
Plug board, The Tech Model Railroad Club
Pong video game, The Homebrew Computer Club
Popular Electronics, Every Man a God, Every Man a God, The Homebrew Computer Club, Tiny BASIC
Pornographic programs, The Homebrew Computer Club
Porter, Katherine Ann, Life
Portola Institute, Revolt in 2100
Prancing Pony, Life
Priesthood, The Tech Model Railroad Club, The Tech Model Railroad Club
Principles of Electricity and Electronics Applied to Telephone and Telegraph Work, The Midnight Computer Wiring Society
Processor Technology, The Homebrew Computer Club, Tiny BASIC, Tiny BASIC, Tiny BASIC, Tiny BASIC, Woz, Secrets
Program bumming, The Tech Model Railroad Club, The Hacker Ethic, The Wizard and the Princess
Programma, The Wizard and the Princess, The Brotherhood
Programming, The Wizard and the Princess, The Wizard and the Princess, Applefest, Wizard vs. Wizards
    social aspects, The Wizard and the Princess
Programming languages, The Wizard and the Princess
programming languages, Afterword: 2010
Project MAC, Spacewar, Greenblatt and Gosper, Greenblatt and Gosper, Winners and Losers, Afterword: 2010
Project One, Revolt in 2100
Propaganda of the Deed, The Homebrew Computer Club

Q

Quinn, Mike, Every Man a God

R

Radio Amateur's Handbook, Revolt in 2100
Radio Electronics, Every Man a God, Every Man a God
Radio Frequency (RP) Modulators, Woz
Radio Shack, Secrets
Rand Corporation, Greenblatt and Gosper
Raster Blaster game, Summer Camp
Red Death, Greenblatt and Gosper

Registers, Greenblatt and Gosper
Reiling, Bob, Woz
Research Laboratory of Electronics (RLE), The Tech Model Railroad Club, Spacewar
Resource One, Revolt in 2100
Revolt in 2100 (Heinlein), Revolt in 2100, Revolt in 2100, Revolt in 2100
Rich and Rich Synergistic Enterprises, Wizard vs. Wizards
Right Thing, The, Greenblatt and Gosper
Robbins, Harold, The Wizard and the Princess
Roberts, Ed, Every Man a God, Every Man a God, The Homebrew Computer Club, Tiny BASIC, Tiny BASIC, Tiny BASIC, Woz
Robotics, Winners and Losers, Life
Robotron game, Frogger, Wizard vs. Wizards
Rolling Stone, Every Man a God
ROM memory boards, The Homebrew Computer Club
ROM-cartridge-based computers, Wizard vs. Wizards, Wizard vs. Wizards
Royalties, The Third Generation, Frogger
Russell, Steve (Slug), Spacewar, Spacewar, Spacewar

S

S-100 bus, Woz
Sagan, Carl, Life
SAGE defense system, The Tech Model Railroad Club, The Midnight Computer Wiring Society
SAIL, Life
Sales figures, Applefest
Samson, Peter, The Tech Model Railroad Club, The Tech Model Railroad Club, The Hacker Ethic, Spacewar, Spacewar, Greenblatt and Gosper, Greenblatt and Gosper, Life, Every Man a God
San Francisco Applefest, Applefest
Saunders, Bob, The Tech Model Railroad Club, The Tech Model Railroad Club, Spacewar, Spacewar
Saunders, Marge, The Tech Model Railroad Club, The Hacker Ethic
Schwader, Warren, Summer Camp, Wizard vs. Wizards
Scientific American, Life
Scott, Mike, Woz
Screenwriter II, Frogger
Sepulveda, Vic, The Third Generation
SHAFT (Society to Help Abolish FORTRAN Teaching), Revolt in 2100
Shootout-at-El-Cassiopeia, Spacewar
Shuttle LIFE pattern, Life
Sierra On-Line, Frogger, Frogger, Frogger, Applefest, Applefest, Wizard vs. Wizards, Wizard vs. Wizards
(see also)
Sierra Sky Ranch area, Summer Camp
Signals and Power (SyE) Subcommittee, The Tech Model Railroad Club, The Midnight Computer Wiring Society
Silicon Valley, Every Man a God
Silver, David, The Midnight Computer Wiring Society, Winners and Losers, Winners and Losers, Life, Life
Singer, Hal, Tiny BASIC
Sirius Software, The Brotherhood, The Brotherhood, The Brotherhood, Frogger, Applefest, Wizard vs. Wizards
Skeet Shoot game, Summer Camp

Skylab 2, Applefest
Smash-Up game, Summer Camp
Smith, Adam, Afterword: 2010
Smith, E. E. (Doc), Spacewar
SMUT-ROMS, The Homebrew Computer Club
Snyder, Tom, Tiny BASIC
Social engineering, Life
Social issues, The Homebrew Computer Club, The Homebrew Computer Club
Society of Creative Anachronisms, Applefest
Softalk magazine, The Brotherhood, The Third Generation, Summer Camp, Summer Camp, Frogger, Applefest, Applefest, Wizard vs. Wizards
Softporn game, Summer Camp
Softsel Distributors, Frogger
Software Arts, Summer Camp
Software Expo, Frogger
Software flap, Tiny BASIC
Software Superstars, The Wizard and the Princess
Sokol, Dan, The Homebrew Computer Club, Tiny BASIC, Woz, Secrets
Sol computer, Tiny BASIC, Woz
Solomon, Les, Every Man a God, The Homebrew Computer Club, The Homebrew Computer Club, Tiny BASIC, Tiny BASIC
SORDMASTER, Applefest
Southern California Computer Society (SCCS), Tiny BASIC
Southwestern Data company, The Brotherhood
Space Eggs game, The Brotherhood
Space Invaders game, The Brotherhood
Spacewar, Spacewar, Life
Spergel, Marty, The Homebrew Computer Club, The Homebrew Computer Club, Woz
Spiradisk, Applefest
Stallman, Richard (RMS), Afterword: 2010, Afterword: 2010
Stanford Medical Center, Every Man a God
Stanford Research Institute, Life, Life
Stanislaus River trip, The Brotherhood
Star Trek game, Revolt in 2100, Every Man a God
Star Wars (movie), Summer Camp
Stargate machine, Wizard vs. Wizards
Starpath, Frogger
Steel, Summer Camp, Applefest
Steffens, Lincoln, Afterword: 2010
Stephenson, Jeff, Summer Camp, Summer Camp, Wizard vs. Wizards, Wizard vs. Wizards
Stockman, Tom, The Tech Model Railroad Club
Stoneware company, The Brotherhood
Stranger in a Strange Land (Heinlein), Revolt in 2100
Sullivan, Jay, The Wizard and the Princess
Summer Camp, Summer Camp
Sunderland, April, Frogger
Sunderland, Dick, The Wizard and the Princess, Frogger, Frogger, Frogger, Applefest, Wizard vs. Wizards, Wizard vs. Wizards, Wizard vs. Wizards
Sup’r Mod modulators, Woz
Sussman, Gerry, Winners and Losers
Sylvania company, Woz
Synapse Software, Wizard vs. Wizards

Synergy, The Homebrew Computer Club
System, The, The Tech Model Railroad Club
Systems Concepts, Life, Every Man a God
Systems hacking, Winners and Losers
Systems programs, The Tech Model Railroad Club
Systems software, The Wizard and the Princess

T

TA Associates, Frogger
Target game, Tiny BASIC, Woz
Tatum, Tom, Wizard vs. Wizards
Taylor, Robert, Life
Tech Model Railroad Club (TMRC), The Tech Model Railroad Club, Greenblatt and Gosper, Greenblatt and Gosper, Greenblatt and Gosper, The Midnight Computer Wiring Society, The Midnight Computer Wiring Society, Frogger
Tech Square, Greenblatt and Gosper, The Midnight Computer Wiring Society, Life, Life, Revolt in 2100
    war-era security measures, Life
Teenage Mutant Ninja Turtles, Afterword: 2010
Telephone interface board, Secrets
Telephone network fingerprinting, Spacewar
Tempest, Frogger
Ten Little Indians (Christie), The Wizard and the Princess
TENEX, Winners and Losers
Third Generation, Summer Camp
Third Generation hackers, Frogger, Applefest
Thompson, Dan, Wizard vs. Wizards
Thompson, Hunter S., Applefest
Thorlin, Fred, The Third Generation
Thread memory, Afterword: 2010
Threshold game, Summer Camp, Summer Camp
Tiger Toys, Frogger
Time sharing, Spacewar, Winners and Losers, Winners and Losers
Time Zone game, Summer Camp, Summer Camp, Wizard vs. Wizards
Tiny BASIC, Tiny BASIC, Tiny BASIC
Tiny BASIC Journal, Tiny BASIC
Tolkien, J.R.R., Life
Tom Swift Terminal, Every Man a God, Tiny BASIC, Secrets
Tommervik, Al, The Brotherhood, The Third Generation, Applefest, Applefest
Tommervik, Margot, The Brotherhood, Applefest, Wizard vs. Wizards
Tomorrow TV show, Tiny BASIC
Tool (jargon), The Tech Model Railroad Club
Tool Room, The Tech Model Railroad Club, Greenblatt and Gosper, Greenblatt and Gosper
Tools for Conviviality (Illich), Every Man a God, Every Man a God
Tools to Make Tools, The Hacker Ethic, Life
Tools, use of, The Midnight Computer Wiring Society
Tools-to-Make-Tools, The Wizard and the Princess
Torvalds, Linus, Afterword: 2010
Transamerica Corporation, Revolt in 2100
Travolta, John, Wizard vs. Wizards
Tri-Pos, Three-Position Display (Minskytron), Spacewar

TRS-80 home computer, Secrets, The Brotherhood
TV Typewriter, Every Man a God, Woz
Twentieth Century Fox Games, Frogger, Wizard vs. Wizards, Wizard vs. Wizards
Twerps game, Frogger
TX-0 (Tixo), The Tech Model Railroad Club, The Hacker Ethic
Tymshare Company, Revolt in 2100

U

U.S. Navy, Greenblatt and Gosper
Ultima 2 game, Applefest
Ultima game, Applefest
Ulysses and the Golden Fleece game, Summer Camp, Wizard vs. Wizards
Univac, Greenblatt and Gosper, Greenblatt and Gosper
Unused op-codes, Winners and Losers
User definable characters, The Third Generation
UT-3 debugging program, The Tech Model Railroad Club

V

VCS machine, The Brotherhood, The Third Generation, Frogger, Wizard vs. Wizards
Venture capital, Frogger, Applefest
VersaWriter, The Wizard and the Princess
Viaweb, Afterword: 2010
VIC-20 computer, Wizard vs. Wizards
Video display module (VDM), Tiny BASIC
Vietnam War, Life
VisiCalc, Summer Camp
Von Braun, Werner, The Tech Model Railroad Club
Vortex Model program, The Midnight Computer Wiring Society
Vril power, The Homebrew Computer Club

W

Wagner, Bob, The Tech Model Railroad Club, The Hacker Ethic
Wall Street Journal, The, Summer Camp, Wizard vs. Wizards
Wall Wars game, The Wizard and the Princess
WarGames (movie), Secrets
Warner Entertainment conglomerate, The Third Generation
Warren, Jim, The Homebrew Computer Club, Tiny BASIC, Woz, Woz, Woz, The Brotherhood
Watchtower publication, Summer Camp
Weather Underground, Life
Weizenbaum, Joseph, Life
Western Electric College Gift Plan, The Tech Model Railroad Club
Western Electric company, Spacewar
WGBH (Boston public TV channel), The Tech Model Railroad Club
Wheels of Zeus, Afterword: 2010
Whirlwind computer, The Tech Model Railroad Club
Whole Earth Catalog, Revolt in 2100, Revolt in 2100, Every Man a God
Wigginton, Randy, Woz, Woz, Woz, Secrets, Secrets
Williams, John, The Third Generation

Williams, Ken, The Wizard and the Princess, The Wizard and the Princess, The Brotherhood, The Third Generation, The Third Generation, Summer Camp, Summer Camp, Summer Camp, Frogger, Frogger, Frogger, Applefest, Applefest, Applefest, Wizard vs. Wizards, Wizard vs. Wizards, Wizard vs. Wizards, Wizard vs. Wizards, Wizard vs. Wizards, Afterword: 2010
Williams, Larry, The Wizard and the Princess
Williams, Roberta, The Wizard and the Princess, The Wizard and the Princess, The Wizard and the Princess, The Brotherhood, Summer Camp, Frogger, Frogger, Frogger, Wizard vs. Wizards, Afterword: 2010
Winnitude, Winners and Losers
Wizard and the Princess game, The Wizard and the Princess, Summer Camp
Wizard of Id cartoon, Wizard vs. Wizards
Wizard vs. Wizards (TV contest), Wizard vs. Wizards
Women hackers, lack of, Greenblatt and Gosper
Women, hackers' relations with, Greenblatt and Gosper, Summer Camp
Wong, Mr. (Chinese chef), Greenblatt and Gosper, Greenblatt and Gosper
Woods, Donald, Life
Word processing, Frogger, Wizard vs. Wizards
Word word-processing software, Afterword: 2010
World Game, Revolt in 2100
Wozniak, Stephen (Steve, Woz), The Homebrew Computer Club, Woz, Woz, Woz, Secrets, Secrets, The Wizard and the Princess, The Brotherhood, The Brotherhood, Applefest, Applefest, Afterword: 2010

X

XDS-940 computer, Life, Revolt in 2100
Xerox, Life, Revolt in 2100, Applefest

Y

Yates, Bill, Every Man a God, Every Man a God

Z

Ziploc bags, The Brotherhood
Zork game, Summer Camp
Zuckerberg, Mark, Afterword: 2010